Gaining Insight into Determinants of Physical Activity using Bayesian Network Learning

BNAIC BENELEARN 2020

Proceedings

November 19–20, 2020

Universiteit Leiden
The Netherlands

Editors: Lu Cao, Walter Kosters and Jefrey Lijffijt

BNAIC/BeneLearn 2020

Proceedings

Leiden, the Netherlands
November 19–20, 2020

Editors: Lu Cao, Walter Kosters and Jefrey Lijffijt

http://www.bnaic.eu

General Chairs

Mitra Baratchi — Leiden University
Jan N. van Rijn — Leiden University

Publication Chair

Frank Takes — Leiden University

Local Organization

Gerrit-Jan de Bruin
Michael Emmerich
Mischa Hautvast
Jaap van den Herik
Mike Huisman
Matthias König
Anna Louise Latour
Enrico Liscio
Michiel van der Meer
Matthias Müller-Brockhausen
Jayshri Murli
Marloes van der Nat
Aske Plaat
Mike Preuss
Peter van der Putten
Suzan Verberne
Jonathan Vis
Hui Wang

Program Committee

Martin Atzmueller — Tilburg University
Bernard de Baets — Ghent University
Mitra Baratchi — Leiden University
Souhaib Ben Taieb — Université de Mons
Floris Bex — Utrecht University
Hendrik Blockeel — Katholieke Universiteit Leuven
Koen van der Blom — Leiden University
Bart Bogaerts — Vrije Universiteit Brussel
Tibor Bosse — Vrije Universiteit Amsterdam
Bert Bredeweg — University of Amsterdam
Egon L. van den Broek — Utrecht University
Lu Cao — Leiden University
Tom Claassen — Radboud University
Walter Daelemans — University of Antwerp
Mehdi Dastani — Utrecht University
Kurt Driessens — Maastricht University
Tim van Erven — Leiden University
Ad Feelders — Utrecht University
George H. L. Fletcher — Eindhoven University of Technology
Benoît Frénay — Université de Namur
Lieke Gelderloos — Tilburg University
Pierre Geurts — University of Liège
Nicolas Gillis — Université de Mons

Nick Harley — Vrije Universiteit Brussel
Frank van Harmelen — Vrije Universiteit Amsterdam
Andrew Hendrickson — Tilburg University
Tom Heskes — Radboud University
Arjen Hommersom — Open University of the Netherlands
Mark Hoogendoorn — Vrije Universiteit Amsterdam
Walter Kosters — Leiden University
Johan Kwisthout — Radboud University
Bertrand Lebichot — Université Libre de Bruxelles
John Lee — Université Catholique de Louvain
Jan Lemeire — Vrije Universiteit Brussel
Tom Lenaerts — Université Libre de Bruxelles
Jefrey Lijffijt — Ghent University
Gilles Louppe — University of Liège
Peter Lucas — Leiden University
Bernd Ludwig — University Regensburg
Elena Marchiori — Radboud University
Wannes Meert — Katholieke Universiteit Leuven
Vlado Menkovski — Eindhoven University of Technology
John-Jules Meyer — Utrecht University
Arno Moonens — Vrije Universiteit Brussel
Nanne van Noord — University of Amsterdam
Frans Oliehoek — Delft University of Technology
Aske Plaat — Leiden University
Eric Postma — Tilburg University
Henry Prakken — University of Utrecht and University of Groningen
Mike Preuss — Leiden University
Peter van der Putten — Leiden University and Pegasystems
Jan N. van Rijn — Leiden University
Yvan Saeys — Ghent University
Chiara F. Sironi — Maastricht University
Evgueni Smirnov — Maastricht University
Gerasimos Spanakis — Maastricht University
Jennifer Spenader — University of Groningen, AI
Johan Suykens — Katholieke Universiteit Leuven
Frank Takes — Leiden University and University of Amsterdam
Dirk Thierens — Utrecht University
Leon van der Torre — University of Luxembourg
Remco Veltkamp — Utrecht University
Joost Vennekens — Katholieke Universiteit Leuven
Arnoud Visser — University of Amsterdam
Marieke van Vugt — University of Groningen
Willem Waegeman — Ghent University
Hui Wang — Leiden University
Gerhard Weiss — University Maastricht
Marco Wiering — University of Groningen
Jef Wijsen — Université de Mons
Mark H. M. Winands — Maastricht University
Marcel Worring — University of Amsterdam
Menno van Zaanen — South African Centre for Digital Language Resources
Yingqian Zhang — Eindhoven University of Technology

Preface

In 2020, the 32nd edition of BNAIC—the annual Benelux Conference on Artificial Intelligence—is organized together with the 29th edition of BeneLearn—the annual Belgian-Dutch Conference on Machine Learning—by the Leiden Institute of Advanced Computer Science (LIACS) of Leiden University, under the auspices of the Benelux Association for Artificial Intelligence (BNVKI).

The conference was scheduled to take place in Corpus, Leiden, but due to the corona virus pandemic and limitations on the organization of events, the conference was organized fully online, for the first time in its history. It took place on Thursday, November 19 and Friday, November 20, 2020. The conference included keynotes by invited speakers, so-called FACt talks, research presentations, a social programme, and a "society and business" afternoon.

The three keynote speakers at the conference were:

- Joost Batenburg, Leiden University
  Challenges in real-time 3D imaging, and how machine learning comes to the rescue
- Gabriele Gramelsberger, RWTH Aachen University
  Machine learning-based research strategies — A game changer for science?
- Tom Schaul, Google DeepMind, London
  The allure and the challenges of deep reinforcement learning

Three FACt talks (FACulty focusing on the FACts of Artificial Intelligence) were scheduled:

- Luc De Raedt, Katholieke Universiteit Leuven
  Neuro-Symbolic = Neural + Logical + Probabilistic
- Nico Roos, Maastricht University
  We aren't doing AI research
- Yingqian Zhang, Eindhoven University of Technology
  AI for industrial decision-making

Authors were invited to submit papers on all aspects of Artificial Intelligence. This year we have received 83 submissions in total. Of the 41 submitted Type A regular papers, both short and long, 24 (59%) were accepted for presentation. All 19 submitted Type B compressed contributions were accepted for presentation. From the Type C demonstrations, 2 out of 3 were accepted. Of the submitted 20 Type D thesis abstracts, 17 were accepted for presentation. Together there are 38 accepted contributions from Type B, C and D. The selection was made based on a single-blind peer review process. Each submission was assigned to three members of the program committee, and their expert reviews were the basis for our decisions. We would like to thank all program committee members (listed on the previous pages) for their time and effort to help us with this task.

All accepted submissions appear in these electronic proceedings, and are made available on the conference web site during the conference. The 12 best accepted regular papers are invited to the postproceedings, to be published in the Springer CCIS series after the conference.

We are grateful to our sponsors for their generous support of the conference:

- SIKS: Netherlands research school for Information and Knowledge Systems
- SNN Adaptive Intelligence: Dutch Foundation for Neural Networks
- BNVKI: Benelux Association for Artificial Intelligence
- SKBS: Stichting Knowledge Based Systems
- ZyLAB
- LIACS: Leiden Institute of Advanced Computer Science

Finally, we would like to thank all who contributed to the success of BNAIC/BeneLearn 2020.

Lu Cao, Walter Kosters and Jefrey Lijffijt
Program Chairs

Contents

Regular papers

Compressed contributions

Regular papers

Evaluating the Robustness of Question-Answering Models to Paraphrased Questions

Paulo Alting von Geusau[0000−0002−3189−4380] and
Peter Bloem[0000−0002−0189−5817]

Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, Netherlands
p.geusau@gmail.com
vu@peterbloem.nl

Abstract. Understanding questions expressed in natural language is a fundamental challenge studied under different applications such as question answering (QA). We explore whether recent state-of-the-art models are capable of recognising two paraphrased questions using unsupervised learning. Firstly, we test QA models' performance on an existing paraphrased dataset (Dev-Para). Secondly, we create a new annotated paraphrased evaluation set (Para-SQuAD) containing multiple paraphrased question pairs from the SQuAD dataset. We describe qualitative investigations on these models and how they present paraphrased questions in continuous space. The results demonstrate that the paraphrased dataset confuses the QA models and leads to a decrease in their performance. Visualizing the sentence embeddings of Para-SQuAD by the QA models suggests that all models, except BERT, struggle to recognise paraphrased questions effectively.

Keywords: natural language · transformers · question answering · embeddings.

1 Introduction

Question answering (QA) is a challenging research topic. Small variations in semantically similar questions may confuse the QA models and result in giving different answers. For example, the questions "Who founded IBM?" and "Who created the company IBM?" should be recognised as having the same meaning by a QA model. QA models need to understand the meaning behind the words and their relationships. Those words can be ambiguous, implicit, and highly contextual.

The motivation for writing this paper springs from the observation that QA models can provide a wrong answer to a question that is phrased slightly different compared to a previous question. Despite the questions being semantically similar. This sensitivity to question paraphrases needs to be improved to provide more robust QA models. Modern QA models need to recognise paraphrases effectively and provide the same answers to paraphrased questions.

Despite the release of high-quality QA datasets, test sets are typically a random subset of the whole dataset, following the same distribution as the development and training sets. We need datasets to test the QA models' ability to recognise paraphrased questions and analyse their performance. Therefore, we use two datasets, based on SQuAD

(Rajpurkar et al., 2016), to conduct two separate experiments on BERT (Devlin et al., 2018), GPT-2 (Radford et al., 2019) and XLNet (Zhilin Yang et al., 2019).

The first dataset we use is an existing paraphrased test set (Dev-Para). Dev-Para is publicly available, and we use it to evaluate the models' over-sensitivity to paraphrased questions.[1] Dev-Para is created from SQuAD development questions and consists of newly generated paraphrases. Dev-Para evaluates the models' performance on unseen test data to gain a better indication of their generalisation ability. We hypothesise that adding new paraphrases to the test set will result in the models suffering a drop in performance. This paper will search for properties that the models learn in an unsupervised way, as a side effect of the original data, setup, and training objective.

In addition, we introduce a new paraphrased evaluation set (Para-SQuAD) to test the QA models' ability in recognising the semantics of a question in an unsupervised manner. Para-SQuAD is a subset of the SQuAD development set, whereas Dev-Para is much larger and consists of newly added paraphrases. Para-SQuAD consists of question pairs that are semantically similar but have a different syntactic structure. The question pairs are manually annotated and picked from the SQuAD development set. We analyse all sentence embeddings of Para-SQuAD in an embedding space with the help of t-SNE visualisation. For each model, we calculate the average cosine similarity of all question pairs to gain an understanding of the semantic similarity between paraphrased questions.

The contributions of this paper are threefold:

1. We test the QA models' performance on an existing paraphrased test set (Dev-Para) to evaluate their robustness to question paraphrases.
2. We create a new paraphrased evaluation set (Para-SQuAD) that consists of question pairs from the original SQuAD development set, the question pairs are semantically similar but have a different syntactic structure.
3. We create and visualize useful sentence embeddings of Para-SQuAD by the QA models, and calculate the average cosine similarity between the sentence embeddings for each QA model.

2 Methodology

In this section, we describe the models and sentence embeddings used, and we introduce our method to create Para-SQuAD.

2.1 BERT, GPT-2 and XLNet

We use QA models that are based on the transformer architecture from Vaswani et al. (2017). The models have been pre-trained on enormous corpora of unlabelled text, including Books Corpus and Wikipedia, and only require task-specific fine-tuning. The first model we use is Google's BERT. BERT is bidirectional because its self-attention

[1] https://github.com/nusnlp/paraphrasing-squad

layer performs self-attention in both directions; each token in the sentence has self-attention with all other tokens in the sentence. The model learns information from both the left and right sides during the training phase. BERT's input is a sequence of provided tokens, and the output is a sequence of generated vectors. These output vectors are referred to as 'context embeddings' since they contain information about the context of the tokens. BERT uses a stack of transformer encoder blocks and has two self-supervised training objectives: masked language modelling and next-sentence prediction.

The second model used in this paper is OpenAI's GPT-2. GPT-2 is also a transformer model and has a similar architecture to BERT; however, it only handles context on the left and uses masked self-attention. GPT-2 is built using transformer decoder blocks and was trained to predict the next word. The model is auto-regressive, just like Google's XLNet.

XLNet, the third model used in this paper has an alternative technique that brings back the merits of auto-regression while still incorporating the context on both sides. XLNet uses the Transformer-XL as its base architecture. The Transformer-XL extends the transformer architecture by adding recurrence at a segment level. XLNet already achieves impressive results for numerous supervised tasks; however, it is unknown if the model generates useful embeddings for unsupervised tasks. We explore this question further in this paper.

We use the small GPT-2, BERT-Base, and XLNet-Base, all consisting of 12 layers. The larger versions of BERT and XLNet have 24 layers; the larger version of GPT-2 has 36 layers.

2.2 Embeddings

Classic word embeddings are static and word-level; this means that each word receives exactly one pre-computed embedding. Embedding is a method that produces continuous vectors for given discrete variables. Word embeddings have demonstrated to improve various NLP tasks, such as question answering (J. Howard and S. Ruder., 2018). These traditional word embedding methods have several limitations in modelling the contextual awareness effectively. Firstly, they cannot handle polysemy. Secondly, they are unable to grasp a real understanding of a word based on its surrounding context.

Advances in unsupervised pre-training techniques, together with large amounts of data, have improved contextual awareness of models such as BERT, GPT-2, and XL-Net. Contextually aware embeddings are embeddings that not only contain information about the represented word, but also information about the surrounding words. The state-of-the-art transformer models create embeddings that depend on the surrounding context instead of an embedding for a single word.

Sentence embeddings are different from word embeddings in that they provide embeddings for the entire sentence. We aim to extract the numerical representation of a question to encapsulate its meaning. Semantically meaningful means that semantically similar sentences are clustered with each other in vector space.

The network structures of the transformer models compute no independent sentence embeddings. Therefore, we modify and adapt the transformer networks to obtain sentence embeddings that are semantically meaningful and used for visualization. We use

The Broncos took an early lead in Super Bowl 50 and never trailed. Newton was limited by Denver's defense, which sacked him seven times and forced him into three turnovers, including a fumble which they recovered for a touchdown. Denver linebacker Von Miller was named Super Bowl MVP, recording five solo tackles, 2½ sacks, and two forced fumbles.
Who was the Super Bowl 50 MVP?
Ground Truth Answers: Von Miller, Miller

Fig. 1. Example of SQuAD 1.1 development set with context, question, and answers.

QA models that are deep unsupervised language representations. All QA models are pre-trained with unlabelled data.

Feeding individual sentences to the models will result in fixed-size sentence embeddings. A conventional approach to retrieve a fixed size sentence embedding is to average the output layer, also called mean pooling. Another common approach for models like BERT and XLNet is to use the first token (the [CLS] token). In this paper, we use the mean pooling technique to retrieve the fixed-size sentence embeddings.

2.3 SQuAD

To create Para-SQuAD, we use the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016), which consists of over 100.000 natural question and answer sets retrieved from over 500 Wikipedia articles by crowd-workers. The SQuAD dataset is widely used as a popular benchmark for QA models. The QA models take a question and context as input to predict the correct answer. The two metrics used for evaluation are the exact match (EM) and the F1 score. The SQuAD dataset is a closed dataset; this means that the answer to a question exists in the context. Figure 1 illustrates an example from the SQuAD development set.

SQuAD treats the task of question answering as a reading comprehension task where the question refers to a Wikipedia paragraph. The answer to a question has to be a span of the presented context; therefore, the starting token and ending token of the substring is calculated.

2.4 Para-SQuAD

To evaluate the robustness of the models on recognising paraphrased questions, we create a new dataset called Para-SQuAD, using the SQuAD 1.1 development set. The SQuAD development set uses at least two additional answers for each question to make the evaluation more reliable. The human performance score on the SQuAD development set is 80.3% for the exact match, and 90.5% for F1.[2]

The first author manually analysed all the questions inside the SQuAD development set to acquire all paraphrased question pairs used in Para-SQuAD. Humans have

[2] https://rajpurkar.github.io/SQuAD-explorer/

a consistent intuition for "good" paraphrases in general (Liu et al., 2010). To be specific, we consider questions as paraphrases if they yield the same answer and have the same intention. The main criteria for well-written paraphrases are fluency and lexical dissimilarity. Moreover, word substitution is sufficient to count as a paraphrase.

Questions in the SQuAD development set relate to specific Wikipedia paragraphs and are grouped together. We manually select paraphrased question pairs that already exist in the SQuAD development set without creating new questions. This method ensures that Para-SQuAD is a typical subset of the SQuAD development set without inducing dataset bias. Moreover, the data distribution and dataset bias in Para-SQuAD and the SQuAD development set remains identical. Para-SQuAD consists of 700 questions, 350 paraphrased question pairs, and 12 different topic categories.

After paraphrase collection, we performed post-processing to check for any mistakes. The paraphrased questions are checked on English fluency using context-free grammar concepts.[3] We used spaCy[4] to conduct a sanity check after manually collecting all paraphrased questions. SpaCy provides paraphrase similarity scores of the question pairs. SpaCy is an industrial-strength natural language processing tool and receives sentence similarity scores by using word embedding vectors.

Using Para-SQuAD for visualisation has a significant advantage compared to using Dev-Para. Namely, the data distribution of Dev-Para changes after the addition of new sentences. On the contrary, the data distribution of Para-SQuAD remains the same because we do not add new sentences; we only annotate the existing paraphrases in the SQuAD development set.

2.5 Para-SQuAD Sentence Embeddings

We present a proof-of-concept visualization of the models' capability to represent semantically similar sentences closely in vector space. Previous research by Coenen et al. (2019) reveals that much of the semantic information, of BERT and related transformer models, is visible and encoded in a low-dimensional space. Therefore, we map all the paraphrased questions from Para-SQuAD to a sentence embedding space for every pre-trained model. Distance in the vector space can be interpreted roughly as sentence similarity according to the model in question.

We calculate the fixed-length vectors for each question using the Flair framework,[5] with mean pooling, to receive the final token representation. Mean pooling uses the average of all word embeddings to obtain an embedding for the whole sentence.

All transformer models produce 768-dimensional vectors for every question, and t-SNE (Laurens van der Maaten and Geoffrey Hinton, 2008) is applied to transform the high-dimensional space to a low-dimensional space in a local and non-linear way. The dimensionality is first reduced to 50 using Principal Component Analysis (PCA) (Karl Pearson, 1901) to ensure scalability, before feeding into t-SNE.

We use a perplexity of 50 for all models, after tuning the 'perplexity' parameter, to capture the clusters. Perplexity deals with the balance between global and local aspects

[3] https://www.nltk.org/
[4] https://spacy.io/
[5] https://github.com/flairNLP/flair

of the data. We tested diverse perplexity values to ensure robustness. We also explore the traditional word-based model GloVe (Pennington et al., 2014) and compare its sentence embeddings to the state-of-the-art transformer models. We investigate if GloVe captures the nuances of the meaning of sentences more effectively as compared to the transformer models.

3 Results

In this section, we evaluate the two experiments. The first experiment measures the performance of the QA models on Dev-Para. The second experiment visualises the sentence embeddings of Para-SQuAD for each QA model.

3.1 Experiments on QA Models

We conduct experiments on three pre-trained models: BERT, GPT-2, and XLNet. The training code of the models is based on the Hugging Face implementation, which is publicly available.[6] In addition to using the pre-trained models directly, we fine-tuned the models on the SQuAD 1.1 training set. We first measure the performance of the pre-trained models on Dev-Para. Secondly, we use the three pre-trained models and GloVe to visualize the sentence embeddings of Para-SQuAD in an embeddings space. Both experiments are performed in an unsupervised manner.

3.2 Dev-Para Performance

We illustrate the performance of all three pre-trained QA models on Dev-Para. Dev-Para consists of the original set and the paraphrased set. The original set contains more than 1.000 questions from the SQuAD development set; the paraphrased set contains between 2 and 3 generated paraphrased questions for each question from the original set (Wee Chung Gan and Hwee Tou Ng, 2019).

The QA models' performance on Dev-Para is presented in Table 1. Although the original set of Dev-Para is semantically similar to the paraphrased set, we see a drop in performance of all three models. Especially GPT-2 and XLNet are suffering a significant drop in performance.

Model	**EM Score**		**F1 Score**	
	Original	Paraphrased	Original	Paraphrased
BERT	82.2	78.7	89.2	86.2
GPT-2	71.6	62.9	80.4	72.7
XLNet	89.4	82.6	93.7	85.3

Table 1. Performance of the QA models on Dev-Para.

[6] https://github.com/huggingface/transformers

The drop in performance is unexpected since the meaning of the questions did not change between the original set and the paraphrased set of Dev-Para. One possible explanation is that the model is exploiting surface details in the original set that are not reproduced by the protocol used to create Dev-Para. If true, this demonstrates a lack of robustness in the models. Moreover, the added questions could be more complicated, therefore allowing for more variability in the syntactic structure, and those questions for which there are paraphrases are variants of more frequent questions.

3.3 Visualization Para-SQuAD

For the following continuous space exploration of Para-SQuAD, we focus on the BERT, GPT-2, XLNet, and GloVe sentence embeddings. Each point in the space represents a question; the 12 colours in Figure 2-5 represent the different categories. The lines in Figure 6-9 illustrate the distance between the paraphrased question pairs. Figure 6-9 all consist of the same amount of lines; however, some lines are difficult to see if both paraphrased question pairs appear close to each other in the embedding space. Paraphrased question pairs that represent the same location in the embedding space appear as a single dot without lines. As a result, it seems that Figure 6 contains fewer lines compared to figure 8, which is a false assumption.

Using visualization as a key evaluation method has important risks to consider. Relative sizes of clusters cannot be seen in a t-SNE plot as dense clusters are expanded, and spare clusters are shrunk. Furthermore, distances between the separated clusters in the t-SNE plot may mean nothing. Clumps of points in the t-SNE plot might be noise coming from small perplexity values.

The visualization of Para-SQuAD consists of all 350 paraphrased question pairs. We argue that the semantics of the questions occupy different locations in continuous space. This hypothesis is tested qualitatively by manually analysing the t-SNE plots of the models. As a sanity check, all sample points in the plots have been manually analysed with the corresponding sentences to check for mistakes (e.g., wrong colour or pairs).

We explore sample points within clusters to gain relevant insights. If two sample points are far from each other in the plot, it does not necessarily imply that they are far from each other in the embedding space. However, the number of long distances between paraphrased question pairs, coming from different clusters, can reveal information on the robustness of the models to recognise paraphrased question pairs and their semantics.

Figure 2 illustrates that BERT creates clear and distinct clusters for every category; we only observe a few errors. Most paraphrased questions are within the same cluster and close to each other (Figure 6). Therefore, it seems that BERT can capture similar semantic sentences effectively.

GPT-2 has trouble clustering the different categories (Figure 3). After manually analysing the sentences in the different clusters, it seems that GPT-2 offers special attention to the first tokens in the sentence. The paraphrased question pairs are close to each other in vector space if they start with the same token. The starting token is often the ‘question word’ in Para-SQuAD. It seems that GPT-2 organises questions by their structure instead of their semantics.

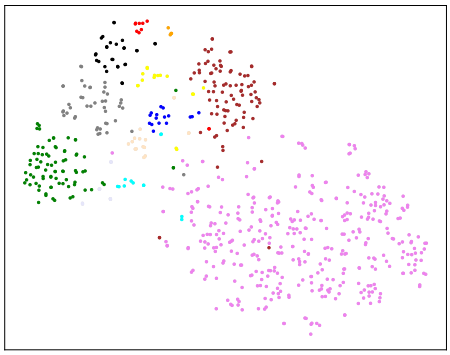


Fig. 2. BERT sentence embeddings.

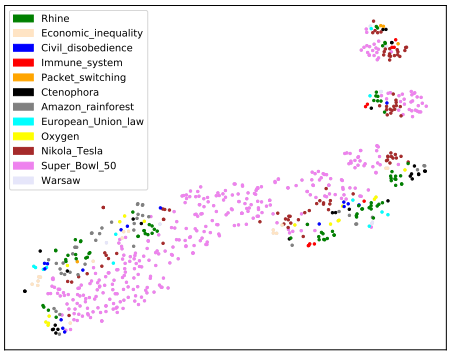


Fig. 3. GPT-2 sentence embeddings.

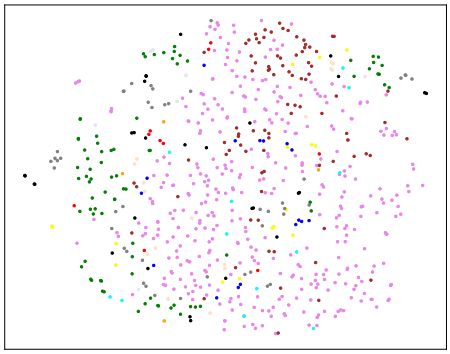


Fig. 4. XLNet sentence embeddings.

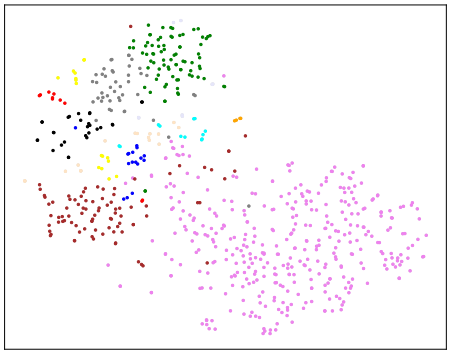


Fig. 5. GloVe sentence embeddings.

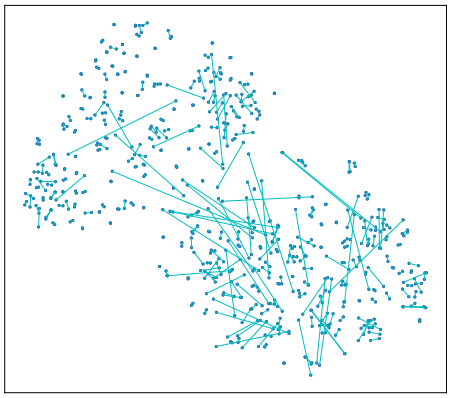


Fig. 6. BERT sentence embeddings.

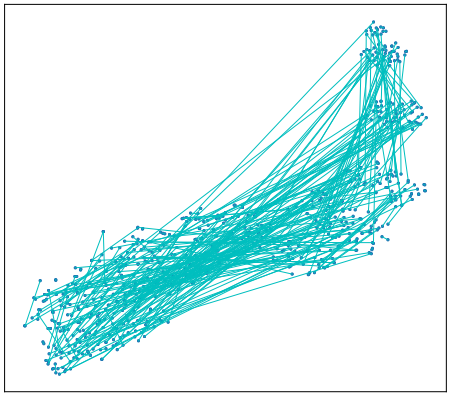


Fig. 7. GPT-2 sentence embeddings.

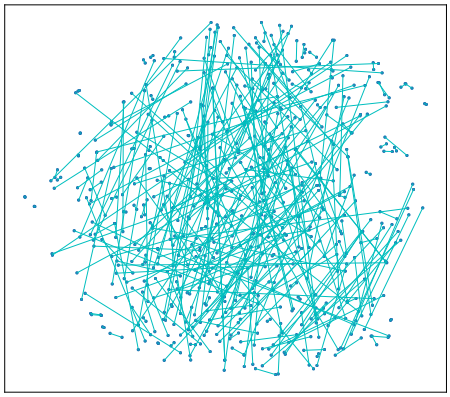


Fig. 8. XLNet sentence embeddings.

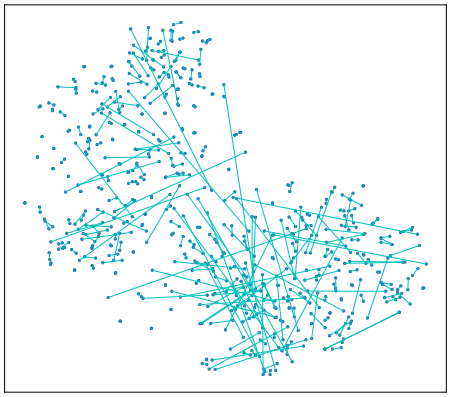


Fig. 9. GloVe sentence embeddings.

XLNet forms one large cluster, with smaller clusters within (Figure 4). However, these clusters are not that clear when compared to BERT. The different categories are all spread out, and no apparent clusters are formed.

Figure 5 suggests that GloVe clusters the different categories more effectively than GPT-2 and XLNet, despite using static embeddings. This finding is interesting, since contextualised embedding are thought to be superior compared to traditional static embeddings. At the same time, the paraphrased questions that appear close to each other in Figure 9 have similar words in the sentence and can be considered as easy paraphrases. GloVe is unable to recognise more complex paraphrases, which can be explained by the model's architecture and not providing contextualised embeddings.

Model	Average Cosine Similarity
BERT	0.875
BERT (fine-tuned)	0.939
GPT-2	0.987
XLNet	0.981

Table 2. Average cosine similarity of the QA models.

In this paper, we use the cosine similarity to measure the closeness between paraphrased question pairs. For each model, we calculate the average cosine similarity for all the paraphrased question pairs in Para-SQuAD to see if the fine-tuned models perform better than the pre-trained models (Table 2). Calculating the average cosine similarity was only relevant for comparing the pre-trained BERT and the fine-tuned BERT. The cosine similarity of the fine-tuned BERT increased with 7.3%. The plots of the fine-tuned models reveal no interesting findings; therefore, we only illustrate the sentence embeddings of the basic pre-trained models.

The average cosine similarity of GPT-2, as illustrated in Table 2, is almost perfect. However, after further investigating the cosine similarity between all paraphrased question pairs, we notice that even two semantically dissimilar sentences have a high cosine similarity. Therefore, this high average reveals extreme anisotropy in the last layers of GPT-2; sentences occupying a tight space in the vector space. We also notice the same effect in XLNet. We can, therefore, suggest that GPT-2 and XLNet are the most context-specific models. This observation is in line with the work of Kawin Ethayarajh (2019).

4 Related Work

Recent research on deep language models and transformer architectures (Vaswani et al., 2017) has demonstrated that context embeddings in transformer models contain sufficient information to perform various NLP tasks with simple classifiers, such as question answering (Tenney et al., 2019; Peters et al., 2018). They suggest that these models produce valuable representations of both syntactic and semantic information.

Attention matrices can encode significant connections between words in a sentence, as illustrated with qualitative and visualization-based work by Jesse Vig (2019). Multiple tests to measure how effective word embeddings capture syntactic and semantic information is defined in the work of Mikolov et al. (2013). Furthermore, the recent work of Hewitt et al. (2019) analysed context embeddings for specific transformer models.

Sentence embeddings can be helpful in multiple ways, analogous to word embeddings. Common proposed methods are: InferSent (Conneau et al., 2017), Skip-Thought (Kiros et al., 2015) and Universal Sentence Encoder (USE) (Cer et al., 2018). Hill et al. (2016) prove that training sentence embeddings on a specific task, such as question answering, impact their quality significantly.

Conneau et al. (2018) presented probing tasks to evaluate sentence embeddings intrinsically. Evaluation of sentence embeddings happens most often in 'transfer learning' tasks, e.g., question type prediction tasks. The study measures to what degree linguistic features, like word order or sentence length, are accessible in a sentence embedding. This study was continued with SentEval (Alexis Conneau and Douwe Kiela, 2018), which serves as a toolkit to evaluate the quality of sentence embeddings. This quality is measured both intrinsically and extrinsically. SentEval proves that no sentence embedding technique is flawless across all tasks (Perone et al., 2018).

Recently, numerous QA datasets have been published (e.g., Rajpurkar et al., 2016; Rajpurkar et al., 2018). However, defining a suitable QA task and developing methodologies for annotation and evolution is still challenging (Kwiatkowski et al., 2019). Key issues include the metrics used for evaluation and the methods and sources used to obtain the questions.

Our analysis focuses on three specific transformer models; however, there are numerous transformer models available. Other notable transformer models are XLM (Lample et al., 2019) and ELECTRA (Clark et al., 2020). Recent papers have focused on generalisability by evaluating different models on several datasets (Priyanka Sen and Amir Saffari, 2020), but not for paraphrasing specifically.

5 Conclusion

This paper presents an initial exploration of how QA models handle paraphrased questions. We used two different datasets and performed tests on each dataset. Firstly, we used an existing paraphrased test set (Dev-Para) to test the QA models' robustness to paraphrased questions. The results demonstrate that all three QA models drop in performance when exposed to more unseen paraphrased questions. The drop in performance could be explained by exposing the models to new paraphrased questions that deviate from the original SQuAD questions. The experiments underline the importance of improving QA models' robustness to question paraphrasing to generalise effectively. Moreover, increased robustness is necessary to increase the reliability and consistency of the QA models when tested on unseen questions in real-life world applications.

Secondly, we constructed a paraphrased evaluation set (Para-SQuAD) based on SQuAD to illustrate interesting insights into QA models handling paraphrased questions. The findings reveal that BERT creates the most promising and informative sentence embeddings and seems to capture semantic information effectively. The other

models, however, seem to fail in recognising paraphrased question pairs effectively and lack robustness.

5.1 Discussion

The models' drop in performance on Dev-Para is unexpected. We hypothesise that the original SQuAD training set does not consist of enough diverse question paraphrases. This lack of variation leads to the QA models not learning to answer different questions, that have the same intention and meaning, correctly. The QA models fail to recognise some questions that convey the same meaning using different wording. Exposing the QA models to more different question phrases would be a logical step to improve the QA models' robustness to question paraphrasing.

Generating paraphrases and recognizing paraphrases are still critical challenges across multiple NLP tasks, including question answering and semantic parsing. A relatively robust and diverse source for generating paraphrases is through neural machine translation. We can make larger datasets consisting of paraphrased questions with the help of machine translation: the question is translated into a foreign language and then back-translated into English. This back-translation approach achieved remarkable results in diversity compared to paraphrases created by human experts (Federmann et al., 2019).

5.2 Limitations

One limitation of the performed experiments is the small size of Para-SQuAD. Increasing Para-SQuAD with data augmentation could be achieved with the use of neural machine translation to generate more paraphrases. Increasing the size of Para-SQuAD would lead to more reliable results, but we would lose the advantage of keeping the data distribution intact.

Another downside is the simplicity of Para-SQuAD. The paraphrases used are relatively simple and basic. Therefore, models achieving excellent results on the set does not guarantee their robustness to question paraphrases.

In general, there is no inter-annotator agreement measure to ensure consistent annotations because we only have one annotator. However, we consider this justified due to the simple task of selecting paraphrased question pairs in the SQuAD development set.

Using visualization as the primary evaluation method has its risks. A common pitfall includes pareidolia; to see structures and patterns that we would like to see. As an example, we can see that BERT forms clear clusters that are known to us; however, other models could form divergent cluster structures to represent patterns. We could, therefore, easily overlook those cluster structures that are unfamiliar to us. Furthermore, clusters can disappear in the t-SNE transformation.

Lastly, with the performed method, it is hard to distinguish whether BERT recognizes the actual semantics of the questions or merely the Wikipedia extracts. Further research is needed to investigate this distinction.

Acknowledgment

We thank the three anonymous reviewers for their constructive comments, and Michael Cochez for his feedback and helpful notes on the manuscript.

References

1. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *arXiv preprint arXiv:1803.11175.*
2. Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *arXiv preprint arXiv:2003.10555.*
3. Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg. 2019. Visualizing and Measuring the Geometry of BERT. *arXiv preprint arXiv:1906.02715.*
4. Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. *arXiv preprint arXiv:1803.05449.*
5. Alexis Conneau, Douwe Kiela, Holger Schwenk, Loıc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.
6. Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *CoRR*, abs/1805.01070.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805.*
8. Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. *arXiv preprint arXiv:1909.00512.*
9. Christian Federmann, Oussama Elachqar, Chris Quirk. 2019. Multilingual Whispers: Generating Paraphrases with Translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019).* Association for Computational Linguistics.
10. Wee Chung Gan and Hwee Tou Ng. 2019. Improving the Robustness of Question Answering Systems to Question Paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
11. John Hewitt and Christopher D Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. *Association for Computational Linguistics.*
12. Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pages 1367– 1377, San Diego, California. Association for Computational Linguistics.
13. J. Howard and S. Ruder. 2018. Fine-tuned Language Models for Text Classification. *CoRR*, abs/1801.06146.
14. Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

15. Tom Kwiatkowski, Jennimaria Palomaki, Olivia Rhinehart, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. In *Transactions of the Association of Computational Linguistics.*
16. Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *arXiv preprint arXiv:1901.07291.*
17. Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng 2010. PEM: A Paraphrase Evaluation Metric Exploiting Parallel Texts. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 923-932.
18. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research,* 9:2579–2605.
19. Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781.*
20. Karl Pearson F.R.S. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* Volume 2.
21. J. Pennington, R. Socher, and C. D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
22. Christian S. Perone, Roberto Silveira, and Thomas S. Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *CoRR*, abs/1806.06259.
23. Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Ken- ton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. *arXiv preprint arXiv:1802.05365.*
24. Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 784–789.
25. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* pages 2383–2392.
26. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog.*
27. Priyanka Sen, Amir Saffari. 2020. What do Models Learn from Question Answering Datasets? *arXiv preprint arXiv:2004.03490.*
28. Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316.*
29. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. *arXiv preprint arXiv:1706.03762.*
30. Jesse Vig. 2019. Visualizing Attention in Transformer-Based Language Representation Models. *arXiv preprint arXiv:1904.02679.*
31. Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237.*

FlipOut: Uncovering Redundant Weights via Sign Flipping*

Andrei C. Apostol[1,2,3], Maarten C. Stol[2], and Patrick Forré[1]

[1] Informatics Institute, University of Amsterdam, The Netherlands
[2] BrainCreators B.V., Amsterdam, The Netherlands
[3] apostol.andrei@braincreators.com

Abstract. We propose a novel pruning method which uses the oscillations around 0 (i.e. sign flips) that a weight has undergone during training in order to determine its saliency. Our method can perform pruning before the network has converged, requires little tuning effort due to having good default values for its hyperparameters, and can directly target the level of sparsity desired by the user. Our experiments, performed on a variety of object classification architectures, show that it is competitive with existing methods and achieves state-of-the-art performance for levels of sparsity of 99.6% and above for 2 out of 3 of the architectures tested. For reproducibility, we release our code publicly at https://github.com/AndreiXYZ/flipout.

Keywords: deep learning · network pruning · computer vision.

1 Introduction

The success of deep learning is motivated by competitive results on a wide range of tasks ([3,9,24]). However, well-performing neural networks often come with the drawback of a large number of parameters, which increases the computational and memory requirements for training and inference. This poses a challenge for deployment on embedded devices, which are often resource-constrained, as well as for use in time sensitive applications, such as autonomous driving or crowd monitoring. Moreover, costs and carbon dioxide emissions associated with training these large networks have reached alarming rates ([21]). To this end, pruning has been proven as an effective way of making neural networks run more efficiently ([5,6,13,15,18]).

Early works ([6,13]) have focused on using the second-order derivative to detect which weights to remove with minimal impact on performance. However, these methods either require strong assumptions about the properties of the Hessian, which are typically violated in practice, or are intractable to run on modern neural networks due to the computations involved.

One could instead prune the weights whose optimum lies at or close to 0 anyway. Building on this idea, the authors of [5] propose training a network until

* Supported by BrainCreators B.V.

convergence, pruning the weights whose magnitudes are below a set threshold, and allowing the network to re-train, a process which can be repeated iteratively. This method is improved on in [4], whereby the authors additionally reset the remaining weights to their values at initialization after a pruning step. Yet, these methods require re-training the network until convergence multiple times, which can be a time consuming process.

Recent alternatives either rely on methods typically used for regularization ([17,18,26]) or introduce a learnable threshold, below which all weights are pruned ([16]). All these methods, however, require extensive hyperparameter tuning in order to obtain a favorable accuracy-sparsity trade-off. Moreover, the final sparsity of the resulting network cannot be predicted given a particular choice of these hyperparameters. These two issues often translate into the fact that the practitioner has to run these methods multiple times when applying them to novel tasks.

To summarize, we have seen that the pruning methods presented so far suffer from one or more of the following problems: (1) computational intractability, (2) having to train the network to convergence multiple times, (3) requiring extensive hyperparameter tuning for optimal performance and (4) inability to target a specific final sparsity.

We note that by using a heuristic in order to determine during training whether a weight has a locally optimal value of low magnitude, pruning can be performed before the network reaches convergence, unlike the method proposed by the authors of [5]. We propose one such heuristic, coined *the aim test*, which determines whether a value represents a local optimum for a weight by monitoring the number of times that weight oscillates around it during training, while also taking into account the distance between the two. We then show that this can be applied to network pruning by applying this test at the value of 0 for all weights simultaneously, and framing it as a saliency criterion. By design, our method is tractable, allows the user to select a specific level of sparsity and can be applied during training.

Our experiments, conducted on a variety of object classification architectures, indicate that it is competitive with respect to relevant pruning methods from literature, and can outperform them for sparsity levels of 99.6% and above. Moreover, we empirically show that our method has default hyperparameter settings which consistently generate near optimal results, easing the burden of tuning.

2 Method

2.1 Motivation

Mini-batch stochastic gradient descent ([2]) is the most commonly used optimization method in machine learning. Given a mini-batch of B randomly sampled training examples consisting of pairs of features and labels $\{(x_b, y_b)\}_{b=1}^{B}$, a neural network parameterised by a weight vector $\boldsymbol{\theta}$, a loss objective $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ and a

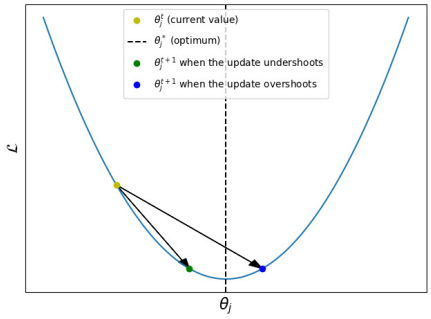


Fig. 1: Over- and under-shooting illustrated. The vertical line splits the x-axis into two regions relative to the (locally-)optimal value θ_j^*. Overshooting corresponds to when a weight gets updated such that its new value lies in the opposite region (blue dot), while undershooting occurs when the updated value is closer to the optimal value, but stays in the same region (green dot).

learning rate η, the update rule of stochastic gradient descent is as follows:

$$\boldsymbol{g}^t = \frac{1}{B}\sum_{b=1}^{B} \nabla_{\boldsymbol{\theta}^t} \mathcal{L}(\boldsymbol{\theta}^t, x_b, y_b)$$
$$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t - \eta \boldsymbol{g}^t$$

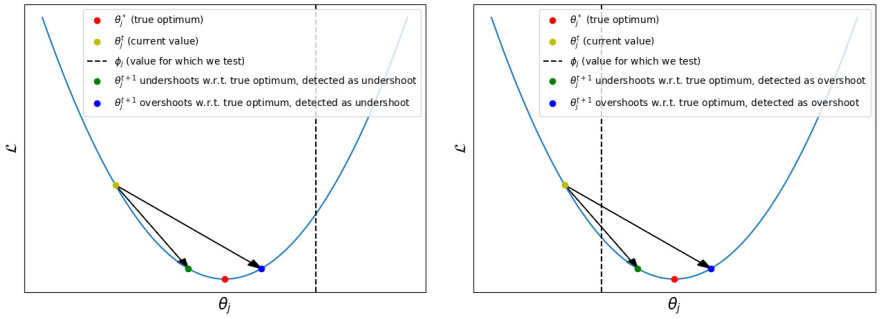
Given a weight θ_j^t, one could consider its possible values as being split into two regions, with a locally optimal value θ_j^* as the separation point. Depending on the value of the gradient and the learning rate, the updated weight θ_j^{t+1} will lie in one of the two regions. That is, it will either get closer to its optimal value while remaining in the same region as before or it will be updated past it and land in the opposite region. We term these two phenomena under- and over-shooting, and provide an illustration in Fig. 1. Mathematically, they correspond to $\eta|g_j^t| < |\theta_j^t - \theta_j^*|$ and $\eta|g_j^t| > |\theta_j^t - \theta_j^*|$, respectively.

With the behavior of under- and over-shooting, one could construct a heuristic-based test in order to evaluate whether a weight has a local optimum at a specific point without needing the network to have reached convergence:

1. For a weight θ_j, a value of ϕ_j is chosen for which the test is conducted
2. Train the model regularly and record the occurrence of under- and over-shooting around ϕ_j after each step of SGD
3. If the number of such occurrences exceeds a threshold κ, conclude that θ_j has a local optimum at ϕ_j (i.e. $\theta_j^* = \phi_j$)

We coin this method *the aim test*.

Previous works have demonstrated that neural networks can tolerate high levels of sparsity with negligible deterioration in performance ([4,5,16,18]). It is then reasonable to assume that for a large number of weights, there exist local



(a) Deceitful observations of under-shooting.(b) Deceitful observations of over-shooting.

Fig. 2: In the plots above, the dotted vertical line represents the value at which the aim test is conducted (i.e. a value we would like to determine as a local optimum or not), while the red dot represents the value of a true local optimum. When testing for a value which is not a locally optimal value $\phi_j \neq \theta_j^*$, over- or under-shooting around ϕ_j can be merely a side-effect of that weight getting updated towards its true optimum θ_j^*. These observations would then contribute towards the aim test returning a false positive outcome, i.e. $\phi_j = \theta_j^*$. Whether we observe an over-shoot or an under-shoot in this case depends on the relationship between ϕ_j and θ_j^*. In (a), we have $\phi_j > \theta_j^*$, where if the hypothesised and true optimum are sufficiently far apart, we observe an under-shoot. Conversely, in (b), we have $\phi_j < \theta_j^*$ and observe over-shooting.

optima at exactly 0, i.e. $\theta_j^* = 0$. One could then use the aim test to detect these weights and prune them. Importantly, when using the aim test for $\phi_j = 0$, the two regions around the tested value are the set of negative and positive real numbers, respectively. Checking for over-shooting then becomes equivalent to testing whether the sign of θ_j has changed after a step of SGD, while under-shooting can be detected when a weight has been updated to a smaller absolute value and retained its sign, i.e. $(|\theta_j^{t+1}| < |\theta_j^t|) \wedge (\text{sgn}(\theta_j^t) = \text{sgn}(\theta_j^{t+1}))$.

However, under-shooting can be problematic; for instance, a weight could be updated to a lower magnitude, while at the same time being far from 0. This can happen when a weight is approaching a non-zero local optimum, an occurrence which should not contribute towards a positive outcome of the aim test. By positive outcome, we refer to determining that $\phi_j = 0$ is indeed a local optimum of θ_j. A similar problem can occur for over-shooting, where a weight receives a large update that causes it to change its sign but not lie in the vicinity of 0. These scenarios, which we will refer to as *deceitful shots* going forward, are illustrated in the general case, where ϕ_j can take any value, in Fig. 2a and Fig. 2b. Following, we make two observations which help circumvent this problem.

Firstly, one could reduce the impact of deceitful shots by also taking into account the distance of the weight to the hypothesised local optimum, i.e. $|\theta_j - \phi_j|$, when conducting the aim test. In other words, the number of occurrences of under-

and over-shooting should be weighed inversely proportional to this quantity, even if they would otherwise exceed κ.

Our second observation is that by ignoring updates which are not in the vicinity of ϕ_j, the number of deceitful shots are reduced. In doing so, one could also simplify the aim test; with a sufficiently large perturbation to θ_j, an update that might otherwise cause under-shooting can be made to cause over-shooting. Adding a perturbation of $\pm\epsilon$ is, in effect, inducing a boundary around the tested value, $[\phi_j - \epsilon, \phi_j + \epsilon]$; all weights that get updated such that they fall into that boundary will be said to over-shoot around ϕ_j. With this framework, checking for over-shooting is sufficient; updates that under-shoot and are within ϵ of the tested value are made to over-shoot (Fig. 3a) and updates which under-shoot but are not in the vicinity of ϕ_j, i.e. a deceitful shot, are now not recorded at all (Fig. 3b). This can also be seen as restricting the aim test to only operate within a vicinity around ϕ_j.

2.2 FlipOut: applying the aim test for pruning

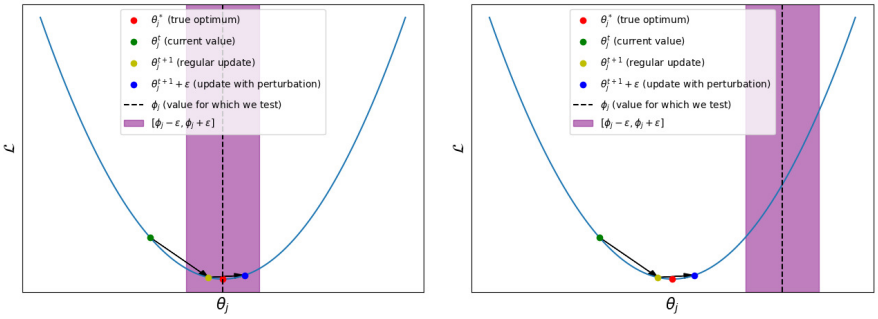
Determining which weights to prune Pruning weights that have local optima at or around 0 can obtain a high level of sparsity with minimal degradation in accuracy. The authors of [5] use the magnitude of the weights once the network is converged as a criterion; that is, the weights with the lowest absolute value (i.e. closest to 0) get pruned. The aim test can be used to detect whether a point represents a local optimum for a weight and can be applied before the network reaches convergence, during training. For pruning, one could then apply the aim test simultaneously for all weights with $\boldsymbol{\phi} = \mathbf{0}$. We propose framing this as a saliency score; at time step t, the saliency τ_j^t of a weight θ_j^t is:

$$\tau_j^t = \frac{|\theta_j^t|^p}{\text{flips}_j^t} \tag{1a}$$

$$\text{flips}_j^t = \sum_{i=0}^{t-1} [\text{sgn}(\theta_j^i) \neq \text{sgn}(\theta_j^{i+1})] \tag{1b}$$

With perturbation added into the weight vector, it is enough to check for over-shooting, which is equivalent to counting the number of sign flips a weight has undergone during the training process when $\phi_j = 0$ (Eq. 1b); a scheme for adding such perturbation is described in Section 2.2. In Equation 1a, the denominator $|\theta_j^t|^p$ represents the proximity of the weight to the hypothesised local optimum, $|\theta_j^t - \phi_j|^p$ (which is equivalent to the weight's magnitude since we have $\phi_j = 0$ for all weights). The hyperparameter p controls how much this quantity is weighted relative to the number of sign flips.

When determining the amount of parameters to be pruned, we adopt the strategy from [4], i.e. pruning a percentage of the remaining weights each time, which allows us to target an exact level of sparsity. Given m, the number of times pruning is performed, r the percentage of remaining weights which are removed at each pruning step, k the total number of training steps, $d_{\boldsymbol{\theta}}$ the dimensionality



(a) Under-shooting can become over-shooting by adding perturbation.

(b) Ignoring deceitful shots.

Fig. 3: (a) All weights that under-shoot but are within ϵ of ϕ_j will be made to over-shoot. (b) When testing at a value which is not a local optimum for θ_j, i.e. $\phi_j \neq \theta_j^*$ and adding a perturbation ϵ to θ_j, not taking under-shooting into account means that if the weight gets updated such that it does not lie in the boundary around ϕ_j induced by the perturbation, an event that would otherwise contribute to a false positive outcome for the aim test will not be recorded, so the likelihood of rejecting ϕ_j as an optimum increases.

of the weights and $||\cdot||_0$ the L_0-norm, the resulting sparsity s of the weight tensor after training the network is simply:

$$s = 1 - \frac{||\boldsymbol{\theta}^k||_0}{d_{\boldsymbol{\theta}}} = (1 - r)^m \tag{2}$$

This final sparsity can then be determined by setting m and r appropriately.

Perturbation through gradient noise Adding gradient noise has been shown to be effective for optimization ([19,25]) in that it can help lower the training loss and reduce overfitting by encouraging an exploration in the parameter space, thus effectively acting as a regularizer. While the benefits of this method are helpful, our motivation for its usage stems from allowing the aim test to be performed in a simpler manner; weights that get updated closer to 0 will occasionally pass over the axis due to the injected noise, thus making checking for over-shooting sufficient. We scale the variance of the noise distribution by the L_2 norm of the parameters $\boldsymbol{\theta}$, normalize it by the number of weights and introduce a hyperparameter λ which scales the amount of noise added into the gradients. For a layer l and d_l its dimensionality, the gradient for the weights in

that layer used by SGD for updates will be:

$$\hat{\boldsymbol{g}}^{t,l} \leftarrow \boldsymbol{g}^{t,l} + \lambda \boldsymbol{\epsilon}^{t,l} \tag{3a}$$

$$\boldsymbol{\epsilon}^{t,l} \sim \mathcal{N}(0, \sigma_{t,l}^2) \tag{3b}$$

$$\sigma_{t,l}^2 = \frac{\|\boldsymbol{\theta}^{t,l}\|_2^2}{d_l} \tag{3c}$$

As training is performed, it is desirable to reduce the amount of added noise so that the network can successfully converge. Previous works use annealing schedules by decaying the variance of the Gaussian distribution proportional to the current time step. Under our proposed formulation, however, explicitly using an annealing schema is not necessary. By pruning weights, the term in the numerator in Eq. 3c decreases, while the denominator remains constant. This ensures that annealing will be induced automatically through the pruning process, and there is no need for manually constructing a schedule.

Pruning periodically throughout training according to the saliency score in Eq. 1a in conjunction with adding gradient noise into the weights forms the *FlipOut* pruning method.

3 Related work

3.1 Deep-R

In Deep-R ([1]), the authors split the weights of the neural network into two matrices, the connection parameter θ_k and a constant sign s_k with $s_k \in \{-1, +1\}$; the final weights of the network are then defined as $\boldsymbol{\theta} \odot \boldsymbol{s}$. The connections whose θ_k is negative are inactive; whenever a connection changes its sign, it is turned dormant and another randomly sampled connection is re-activated, ensuring the same sparsity level is maintained throughout training. Gaussian noise is also injected into the gradients during training.

Two similarities with our method can be observed here, namely the fact that the authors also use sign flipping as a signal for pruning a weight, and the addition of Gaussian noise. However, our methods differ in that we do not impose a set level of sparsity throughout training; instead, we use the number of sign flips of a weight in order to determine its saliency, while in Deep-R a single sign flip is required for a weight to be removed. Our method of injecting noise into the gradients also differs in that it does not explicitly encode an annealing scheme, allowing for the pruning process itself to reduce the noise throughout training. Finally, in Deep-R, the network is initialized with a specific level of sparsity which is maintained throughout training, while our method prunes gradually.

3.2 Magnitude and uncertainty pruning

The M&U pruning criterion is proposed in [11]. Given a weight θ_j, its uncertainty estimate $\tilde{\sigma}_{\theta_j}$ and a parameter λ controlling the trade-off between magnitude and

uncertainty, the M&U criterion will evaluate the saliency of the weight as:

$$\tau_j = \frac{|\theta_j|}{\lambda + \tilde{\sigma}_{\theta_j}}$$

Uncertainty is estimated as the standard deviation across the previous n values of that weight, via a process called pseudo-bootstraping. This criterion is a generalization of the Wald test, and is equivalent to it when $\lambda = 0$.

Our method is similar in that our saliency score also normalizes the weight's magnitude by a function of its past values. However, this method assumes asymptotic normality. While this is the case when using negative log-likelihood or an equivalent as the loss function, this property does not necessarily hold when using modified variants of the SGD estimator, such as Adam ([10]) or RMSprop ([22]). In contrast, FlipOut is not derived from the Wald test and does not make any assumptions about the weight distribution at convergence.

4 Experiments

4.1 General Setup

Baselines As baselines, we consider a slightly modified version of magnitude pruning ([5]) (Global magnitude), due to the similarity between its saliency criterion and that of our own method, SNIP ([14]) due to it being an easily applicable method which does not suffer from any of the issues that are commonly found in pruning methods (Section 1) and Hoyer-Square, as introduced in [26], for the state-of-the-art results that it has demonstrated. We also include random pruning (Random) as a control. For FlipOut, Global magnitude and Random, pruning is performed periodically throughout training. We compare these methods at five different compression ratios, chosen at regular log-intervals (Table 1); for Hoyer-square, the performance at those points is estimated by a sparsity-accuracy trade-off curve. Magnitude pruning, in its original formulation, performs pruning only once the network has reached convergence. However, employing this strategy can create a confounding variable: training time. Since we would like to compare all methods at equal training budgets, we have opted to simply perform pruning after a fixed number of epochs for these methods. Note that the training budget that we allocate allows all of the networks that we consider to reach convergence when trained without performing any pruning. We make an exception to this equal budget rule for Hoyer-Square, since it prunes after training and would otherwise not benefit from any SGD updates after sparsification. As such, we have performed an additional 150 epochs of fine-tuning without the regularizer, as per the original method, although we have observed negligible benefits to this. All baselines were modified to rank the weights globally when a pruning decision is made, as per the strategy from [4], in order to avoid creating bottleneck layers. The models that we test on are ResNet18 ([7]) and VGG19 ([20]) trained on the CIFAR-10 dataset ([12]), and DenseNet121 ([9]) trained on Imagenette ([8]).

Table 1: Compression ratios, resulting sparsity levels and prune frequencies used in the experiments, assuming 350 epochs of training and that 50% of the remaining weights are removed at each step.

Compression ratio ($\frac{d_{\boldsymbol{\theta}}}{\|\|\boldsymbol{\theta}\|\|_0}$)	Resulting sparsity ($1 - \frac{\|\|\boldsymbol{\theta}\|\|_0}{d_{\boldsymbol{\theta}}}$)	Epochs before pruning
2^2	75%	117
2^4	93.75%	70
2^6	98.43%	50
2^8	99.61%	39
2^{10}	99.9%	32

Hyperparameters The training parameters for all experiments are taken from [23]; specifically, we use a learning rate of 0.1, batch size of 128, 350 epochs of training and a weight decay penalty of $5e-4$. The learning rate is decayed by a factor of 10 at epochs 150 and 250. The networks are trained with the SGD optimizer with a momentum value of 0.9 ([2]). For the methods that perform iterative pruning (Global magnitude, Random, FlipOut), we remove 50% of the remaining weights at each pruning step, with the pruning frequencies chosen such that the compression ratios from Table 1 are achieved; we use the same pruning rates and frequencies across all three methods. SNIP accepts a single hyperparameter, namely the desired final sparsity, which we have chosen such that it matches the aforementioned compression ratios. For Hoyer-Square, which does not allow for a specific level of sparsity to be chosen and, instead, relies on parameter tuning, we generate a sparsity-accuracy trade-off curve by using 15 different values for the regularization term, ranging from $1e-7$ to $6e-3$ with 3 values at each decimal point (e.g. $1e-7$, $3e-7$, $6e-7$, $1e-6$ etc.) and a fixed pruning threshold of $1e-4$. Finally, for FlipOut, we use the values of $p=2$ (Eq. 1) and $\lambda=1$ (Eq. 3) for all experiments, a choice we motivate in Section 4.2.

4.2 Choosing the hyperparameters for FlipOut

We have experimented with different values of the two hyperparameters and found that $p=2$ (Eq. 1a) and $\lambda=1$ (Eq. 3a) offer consistent, strong results for all networks tested. In the following paragraphs, we detail the procedure used in determining these values.

Choosing $\boldsymbol{\lambda}$ For λ, we have run all networks at 15 different values, ranging from 0.75 to 1.5 in increments of 0.05. The value of $p=2$ was used. The networks are evaluated on a validation set, created by removing a random subset of samples from the training set. The size of the validation set was 10000 for CIFAR10 and 2000 for Imagenette. For our subsequent experiments, (Sections 4.3 and 4.4), the networks have been trained on the full training set. As a metric, we have used the accuracy of the networks at the end of training for the sparsity levels of 93.75% and 99.9%. We provide in Table 2 the accuracies generated by the optimal

Table 2: Accuracies when using the best value of λ discovered by grid search and the value of $\lambda = 1$ at two levels of sparsity. The parantheses indicate the gain offered by the optimal parameter.

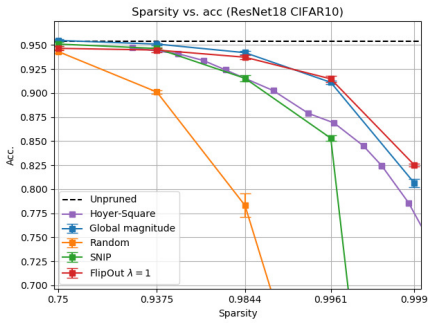
	Acc. at sparsity 93.75%		Acc. at sparsity 99.9%	
Model	λ^*	$\lambda = 1$	λ^*	$\lambda = 1$
ResNet18	94.58(+0.02)	94.56	83.75(+1.68)	82.07
VGG19	93.07(+0.11)	92.96	87.72(+0.48)	87.24
DenseNet121	89.75(+0.0)	89.75	73.5(+1.45)	72.05

Table 3: Table of results for different values of p at two levels of sparsity.

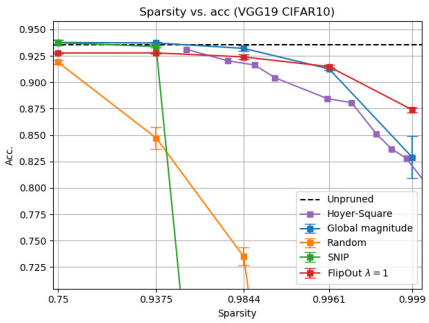
	Acc. at sparsity 93.75%					Acc. at sparsity 99.9%				
Model	$p = 0$	$p = \frac{1}{2}$	$p = 1$	$p = 2$	$p = 4$	$p = 0$	$p = \frac{1}{2}$	$p = 1$	$p = 2$	$p = 4$
ResNet18	93.71	88.39	94.18	**94.26**	94.11	72.69	77.08	79.83	82.07	**83.15**
VGG19	91.68	82.44	92.56	**92.96**	92.57	81.48	80.69	86.01	**87.24**	86.64
DenseNet121	10.35	77.40	88.9	**89.75**	88.86	10.35	10.35	70.85	**72.05**	60.55

value of λ, as discovered through this process, and the ones generated at $\lambda = 1$. Notice that the differences are almost negligible at 93.75% sparsity. For the larger sparsity level the disparity increases, although the default value still remains within 2 percentage points of the optimum value for all networks considered. The largest gap can be seen for ResNet18 and DenseNet121, at approximately 1.7 and 1.5 percentage points, respectively. Since there are only two out of six cases in which optimizing λ has helped beyond a negligible amount, we have used the value of 1 for this hyperparameter throughout our experiments.

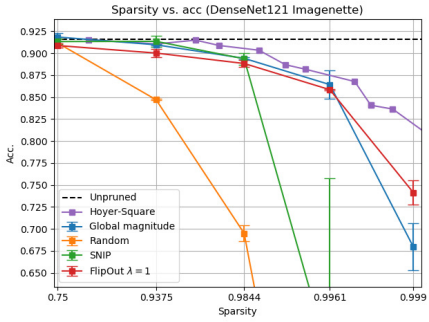
Choosing *p* We perform similar experiments for p on five values, $p \in \{0, \frac{1}{2}, 1, 2, 4\}$. Note that the value of $p = 0$ corresponds to the case when the magnitudes of the weights are not taken into account; that is, the pruning decisions will be made solely based on the number of sign flips. As can be seen in Table 3, the value of $p = 2$ consistently outperforms all other tested values, with the exception of ResNet18 at 99.9% sparsity, for which the value of $p = 4$ achieves better results by approximately 1 percentage point. Another interesting observation is that the values of 1, 2 and 4 tend to perform better than 0 and $\frac{1}{2}$; we conjecture that this is due to the fact that deceitful shots (Section 2.1) occur when not taking into account the distance between the weight and its hypothesised local optimum, which have a negative impact on the pruning decision. This can be especially observed at the higher sparsity level and in the case of DenseNet121, where pruning with $p = 0$ causes the network to not perform better than random guessing. Given that the value of $p = 2$ is favored in 5 out of 6 cases, we have decided to use it as a default value in our subsequent experiments.



(a) ResNet18 on CIFAR10



(b) VGG19 on CIFAR10



(c) DenseNet121 on Imagenette

Fig. 4: Results of pruning ResNet18 and VGG19 on the CIFAR10 dataset. Each point is averaged over 3 runs; error bars indicate standard deviation.

4.3 Comparison to baselines

The results for the three models tested are found in Figure 4. FlipOut obtains state-of-the-art performance on ResNet18 and VGG19 for sparsity levels of 99.61% and beyond. For the highest tested sparsity level, it outperforms the second-best method by 1.9 and 4.5 percentage points, respectively (Fig. 4a, 4b). Notably, when using FlipOut on VGG19 for this sparsity, the drop in accuracy compared to the unpruned model is only 6.2 percentage points. At the same time, it remains competitive with other baselines for lower degrees of sparsity, staying within a 1 percentage point difference compared to the best method and with a minimal drop relative to the unpruned model. For DenseNet121, however, Hoyer-Square dominates all other methods tested in most cases (Fig. 4c), with FlipOut as second best for the highest sparsity level.

Interestingly, the simple criterion of magnitude pruning, when modified to rank the weights globally instead of a layer-by-layer basis, is competitive with other, more recent, baselines, and even obtains state-of-the-art results for moderate levels of sparsity. However, at high levels of sparsity, which correspond to more

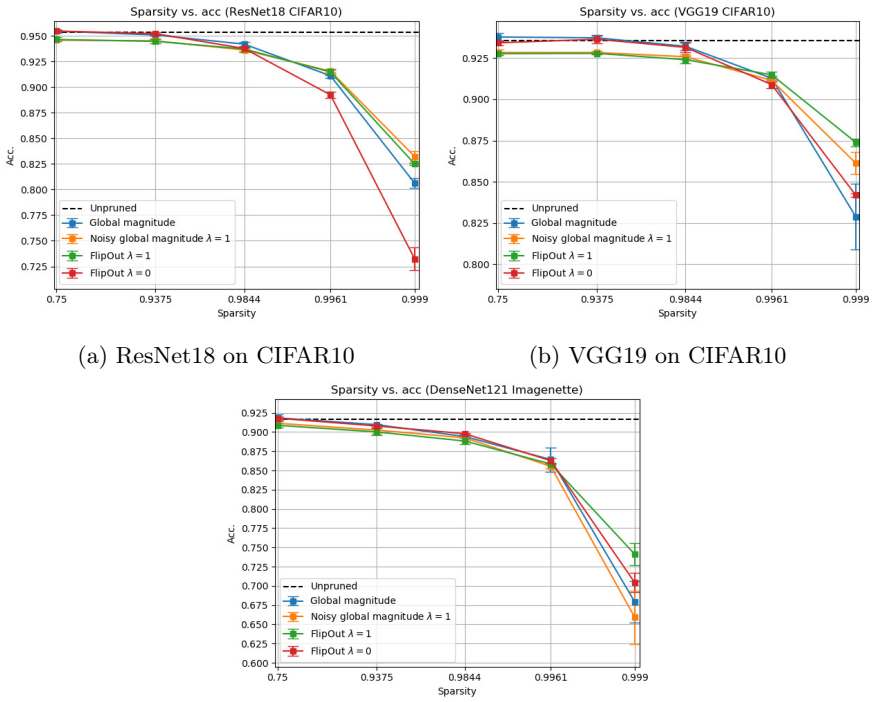
frequent and implicitly earlier pruning steps (Table 1) there is a performance degradation. This suggests that the magnitude of a weight by itself is not a good measure of saliency when the network is far from reaching convergence. It is also worth noting that SNIP collapses at high levels of sparsity, causing the network to perform no better than random guessing. Upon inspecting these cases (not shown for visibility) we noticed that at least one layer has been entirely pruned, effectively blocking any signal from passing. Interestingly, this does not happen for any of the other baselines (except for Random). We conjecture that this collapse as well as the cases where SNIP performs worse than random pruning (Fig. 4b) are a result of pruning at initialization; pruning too too early can cause the saliency criterion to be inaccurate, but also impedes training in and of itself.

During our experiments, we empirically observed that Hoyer-Square requires extensive hyperparameter tuning for optimal performance. Our method, however, has strong default values and can also target the final sparsity directly, while also not requiring additional epochs of fine-tuning. Finally, SNIP, the only other baseline which does not suffer from any of the issues commonly found among pruning methods (Section 1) compromises on performance for high levels of sparsity, whereas FlipOut does not.

4.4 Is it just the noise?

The performance of FlipOut could simply be a result of the noise addition, which is known to aid optimization ([19,25]). To investigate this, we perform experiments with global magnitude as the pruning criterion in which we add noise into the gradients using the recipe from Equation 3*c* and compare it to our own method. Notably, the saliency criterion of these two methods differ only in that FlipOut normalizes the magnitude by the number of sign flips (denominator in Eq. 1a). The hyperparameters were kept at their default values of $p = 2$ for FlipOut and $\lambda = 1$ for both methods. We also include runs of FlipOut where no noise was added (i.e. $\lambda = 0$). These serve as a control, decoupling the two novel components of our method: noise addition and scaling magnitudes by the number of sign flips. The same pruning rates and frequency of pruning steps have been used as before (Table 1). The results are illustrated in Fig. 5.

For sparsity levels up to 98.44%, adding gradient noise causes a slight deterioration on performance, as can be seen by the fact that both global magnitude and FlipOut with $\lambda = 0$ outperform their noisy counterparts. It can also be seen that FlipOut with $\lambda = 1$ performs comparably to noisy global magnitude, indicating that measuring saliency by sign flips does not benefit accuracy in these regimes compared to using only the magnitude, and the performance gap between the noisy and non-noisy methods is likely a result of noise addition. For sparsity levels of 99.61% and above, however, the opposite is true. It seems that gradient noise disproportionately benefits networks with a small number of remaining parameters; we conjecture that this is due to the fact that the exploration in parameter space induced by noise is more effective when that space is heavily constrained. Focusing on the highest level of sparsity, FlipOut outperforms noisy global magnitude on VGG19 (Fig. 5b) and DenseNet121 (Fig. 5c) by 1.2 and



(a) ResNet18 on CIFAR10 (b) VGG19 on CIFAR10

(c) DenseNet121 on Imagenette

Fig. 5: Results of the ablation study on the noise. Each point is averaged over 3 runs. Global magnitude without adding noise is also shown for comparison.

8.2 percentage points, respectively, while being outperformed by 0.8 percentage points on ResNet18 (Fig. 5a). The standard deviation of FlipOut at this point is lower than for noisy global magnitude for all networks tested, making it more robust to initial conditions and the noise sampling process. At this level, the addition of gradient noise to FlipOut also shows performance boosts compared to its non-noisy counterpart, namely 9.3 percentage points for ResNet18, 3.2 for VGG19 and 3.7 for DenseNet121. The benefits caused by adding noise to global magnitude as compared to adding it to FlipOut are similar for VGG19; however, it is relatively small for ResNet18 at 2.6 percentage points and even causes a 2 percentage point drop in performance for DenseNet121.

Since FlipOut with $\lambda = 1$ outperforms noisy global magnitude in 2 out of 3 cases for the highest level of sparsity while maintaining similar performance in all other cases as well as being less sensitive to the choice of seed, we conclude that its results cannot be explained only by the addition of noise and is also caused by the sign flips being taken into account when computing saliency.

Additionally, we conjecture that occurrences of under-shooting are indeed converted into over-shooting when adding gradient noise, allowing FlipOut to more accurately compute saliencies. This is evidenced by the fact that gradient noise addition benefits FlipOut more so than it does global magnitude, and implies that our method of dealing with deceitful shots is sound.

5 Discussion

In this work, we introduce the aim test, a general method for determining whether a point represents a local optimum for a weight during training, and propose using it for pruning by applying the test for all weights simultaneously and framing it as a saliency criterion. This method, coined FlipOut, demonstrates several desirable qualities: it is computationally tractable, allows for an exact level of sparsity to be selected, requires a single training run and has default hyperparameter settings which generate near optimal results, easing the burden of hyperparameter search.

We compare the performance of FlipOut to relevant baselines from literature on a variety of object classification architectures. We show that it achieves state-of-the-art performance at the highest levels of sparsity tested for 2 out of 3 networks, and maintains competitive performance in less sparse regimes. Finally, we conduct an ablation study on the two components of our algorithm, gradient noise addition and the saliency criterion, and find that both play an important role in yielding this performance performance.

References

1. Bellec, G., Kappel, D., Maass, W., Legenstein, R.: Deep rewiring: Training very sparse deep networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=BJ_wN01C-
2. Bottou, L.: Online algorithms and stochastic approximations. In: Saad, D. (ed.) Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK (1998), http://leon.bottou.org/papers/bottou-98x, revised, oct 2012
3. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=B1xsqj09Fm
4. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=rJl-b3RcF7
5. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in neural information processing systems. pp. 1135–1143 (2015)
6. Hassibi, B., Stork, D.G.: Second order derivatives for network pruning: Optimal brain surgeon. In: Advances in neural information processing systems. pp. 164–171 (1993)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

8. Howard, J.: Imagenette (2019), accessed April 6th, 2020 at https://github.com/fastai/imagenette/tree/6395a747bef7a9760b95cd582ece09d90f8a4769
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
10. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2015)
11. Ko, V., Oehmcke, S., Gieseke, F.: Magnitude and uncertainty pruning criterion for neural networks. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 2317–2326 (2019)
12. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
13. LeCun, Y., Denker, J.S., Solla, S.A.: Optimal brain damage. In: Touretzky, D.S. (ed.) Advances in Neural Information Processing Systems 2, pp. 598–605. Morgan-Kaufmann (1990), http://papers.nips.cc/paper/250-optimal-brain-damage.pdf
14. Lee, N., Ajanthan, T., Torr, P.: Snip: Single-shot network pruning based on connection sensitivity. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=B1VZqjAcYX
15. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: International Conference on Learning Representations (2017)
16. Liu, J., Xu, Z., Shi, R., Cheung, R.C.C., So, H.K.: Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=SJlbGJrtDB
17. Louizos, C., Welling, M., Kingma, D.P.: Learning sparse neural networks through l_0 regularization. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=H1Y8hhg0b
18. Molchanov, D., Ashukha, A., Vetrov, D.: Variational dropout sparsifies deep neural networks. In: Proceedings of the 34th International Conference on Machine Learning (2017)
19. Neelakantan, A., Vilnis, L., Le, Q.V., Sutskever, I., Kaiser, L., Kurach, K., Martens, J.: Adding Gradient Noise Improves Learning for Very Deep Networks. arXiv e-prints arXiv:1511.06807 (Nov 2015)
20. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations (2015)
21. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp. In: ACL (2019)
22. Tieleman, T., Hinton, G.: "Lecture 6.5—rmsprop: Divide the gradient by a running average of its recent magnitude". COURSERA: Neural Networks for Machine Learning (2012)
23. Train CIFAR10 with PyTorch (GitHub Repository), Unknown Author: Pytorch cifar-10 github repository (2017), accessed April 6th, 2020 at https://github.com/kuangliu/pytorch-cifar/tree/ab908327d44bf9b1d22cd333a4466e85083d3f21
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
25. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 681–688 (2011)
26. Yang, H., Wen, W., Li, H.: Deephoyer: Learning sparser neural network with differentiable scale-invariant sparsity measures. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=rylBK34FDS

Emotion Intensity and Gender Detection via Speech and Facial Expressions

Elahe Bagheri[1], Oliver Roesler[2], Hoang-Long Cao[1], and Bram Vanderborght[1]

Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel and Flanders Make, Brussels, Belgium.
Artificial Intelligence Lab, Vrije Universiteit Brussel, Brussels, Belgium.
elahe.bagheri@vub.be, oliver@roesler.co.uk, hoang.long.cao@vub.be,
bram.vanderborght@vub.be

Abstract. Human emotion detection has received increasing attention over the last decades for a variety of applications and systems. However, detecting the intensity of the expressed emotion has not been investigated as much as detecting the type of the expressed emotion. To fill this gap, we investigate the utility of different facial and speech features for emotion intensity detection. To this end, we designed different Deep Neural Network based models and applied them to the RAVDESS dataset. Obtained results show that speech signal features are better indicators of emotion intensity than facial features. However, in the absence of speech signals, finding emotion intensity by facial expressions is more accurate for males in comparison to females.
The difference between the accuracy of emotion intensity detection for two genders motivated us to use speech signals for the gender detection task. The obtained results confirm that the proposed model achieves higher accuracy in emotion intensity detection and is more robust in gender detection than the state-of-the-art.

Keywords: Emotion Intensity Detection · Gender Detection · Deep Learning.

1 INTRODUCTION

Detecting human emotions is crucial in developing cognitive and adaptive behaviors for artificial intelligent systems, robots, and (virtual) agents. Emotion detection is the ability to recognize another's affective state, which typically involves the integration and analysis of human expressions through different modalities, like facial expression, speech, body movements, and gestures [5]. Mehrabian [21] showed 55% of human emotions are conveyed through facial expression and 38% through speech, therefore, facial and speech emotion recognition received significant attention during the last decades. Although finding the type of expressed emotion is essential to adapt to a user's affective state, it is not enough, and a difference in intensity has been proven to be important to distinguish different emotional states [13]. For instance, a polite smile versus embarrassed smile [1] and posed versus spontaneous smile are separable by differences in their expression intensities [7]. Since there is not much research on emotion intensity detection, in the following sections, the state-of-the-art in speech emotion recognition and facial emotion recognition are discussed.

1.1 Speech Emotion Recognition (SER)

The effect of emotions can be seen in both acoustic characteristics and lexical content of speech. Some examples of acoustic features are Mel-Frequency Cepstral Coefficients (MFCC), energy, jitter, and shimmer, which are known as Low-Level Descriptors (LLDs) [14] [1]. Some examples of lexical features are the presence/absence of word stems, and bag-of-words sentiment categories [28]. However, when the linguistic content is not emotionally rich, recognizing emotion from the transcript is very difficult [28], thus, in this study, our focus is on the acoustic characteristics for recognizing the emotion intensity.
In traditional SER methods, the acoustic features are first extracted and then different machine learning algorithms like Support Vector Machine (SVM) [23], K-nearest neighbor [15] and Hidden Markov model [26] are applied to the obtained features to classify them into considered emotion classes. To obtain these features from utterance-level, each signal is broken into shorter frames of 20 to 50 milliseconds, and their features, i.e., frame-level features, are extracted. However, emotional contents are not in static values of these features but are in their temporal variations. Thus, different statistical functions, e.g., minimum, maximum, mean, variance, linear regression coefficients, etc., are applied to these

[1] Some other investigated acoustic characteristics, i.e., LLDs, are zero-crossing rate, duration, and higher-order formants, Mel-filterbank features, spectral features, formant locations/bandwidths, perceptual linear prediction, fundamental frequency, and pitch.

features to illustrate their temporal variations and contours. The obtained results, afterward, are unified in a vector to achieve utterance-level features [22].
Due to the success of deep learning in different fields like image, video, and natural language processing, the interest in applying Deep Neural Networks (DNN) for speech emotion recognition also increased. Authors in [11] and [17] used deep feed-forward and recurrent neural networks to learn the frame-level acoustic features, and used extreme learning machines for the utterance-level aggregation. Mirsamadi et al. [22] used Rectified Linear Unit (ReLU) dense layers to learn frame-level features, and Bidirectional Long Short-Term Memory (BiLSTM) recurrent layers to learn the temporal aggregation. Neumann et al. [25] used a Convolutional Neural Network (CNN) with one convolutional layer and one pooling layer, to learn the representation of the audio signal, and an attention layer to compute the weighted sum of all the information extracted from different parts of the input. Lim et al. [18] transformed the speech signal into 2D representation using Short Time Fourier Transform and sent them to concatenated CNN and LSTM architectures without using any traditional hand-crafted features. Trigeorgis et al. [32] applied two BiLSTM layers to balance the frame-level characterization and utterance-level aggregation and transform frame-level convolutional features directly into continuous arousal and valence output so that the model learned direct mapping from time-domain speech signals into the continuous model of emotion. The temporal model proposed in [12] used BiLSTM to represent forward/backward contextual information of temporal dynamics of the speech signal and conducted a CNN and a capsule net to learn temporal clusters and classify the extracted patterns. Mustaqeem et al. [24] obtained the spectrogram of signals and then used CNN and LSTM to classify the speech.

1.2 Facial Emotion Recognition (FER)

Ekman [8] showed six basic emotions[2] are expressed universally the same through facial muscles. He introduced the Action Unit (AU) to indicate fundamental movements of a single or group of muscles through the facial expression of a special emotion [3]. He also defined the Facial Action Coding System (FACS) to encode the movements of these AUs [10]. One way to recognize a facially expressed emotion is detecting the status of all individual AUs and then analyzing the combination of the activated AUs to obtain the expressed emotion. On the other hand, promising results of DNN based approaches in comparison with classical machine learning algorithms lead to the proposal of numerous DNN based FER methods in the research community. For instance, Bagheri et al. [2] used facial muscle activities as raw input for a Stacked Auto Encoder (SAE). The applied SAE returns the best combination of muscles in describing a particular emotion, which is then sent to a Softmax layer to fulfill the multi-classification task. Liu et al [19] proposed a sign-based DNN architecture to investigate the effect of AUs in emotion recognition. The proposed model consists of three sequential modules, where the first module generates a complete representation of all expression-specific appearance variations by a convolution layer stacked by a max-pooling layer. The second module finds the best simulation of the combination of the AUs and the last module learns hierarchical features by Restricted Boltzmann Machines (RBM). Finally, a linear SVM classifier is used to recognize the six basic emotions. However, AU-aware layers, in the second module, are not able to detect all FACS in images. Pitaloka et al [27] used CNN to extract features from an input image, which is then passed to a max-pooling layer to reduce the image size. A fully connected layer, in the end, classifies the input image into one of the six basic emotions. However, the performance of the proposed algorithm decreases when the dimension of the input image increases regarding the complexity of the high dimensional images.

Research on emotion intensity detection has been focused on the estimation of the intensity of Action Units (AU), e.g., [6, 13] and FERA 2015[4], however, there is no conclusion about the intensity of the expressed emotion, thus, the goal of this study is developing a model by which the intensity of the expressed emotion in a given image, speech signal or video can be identified.
The remainder of this paper is structured as follows: the applied models, dataset, and extracted features are explained in Section 2. Section 3 demonstrates the conducted experiments and obtained results. Finally, Section 4 concludes this paper.

[2] Happiness, sadness, fear, anger, surprise, and disgust.
[3] `https://imotions.com/blog/facial-action-coding-system/`
[4] Facial Expression Recognition and Analysis challenge

Table 1
The architectures of the proposed DNN based models.

LSTM		BiLSTM		CNN		
Simple	**Attention**	**Simple**	**Attention**	**Simple**	**BiLSTM/LSTM**	**BiLSTM/LSTM+Attention**
LSTM	LSTM	BiLSTM	BiLSTM	CNN	CNN	CNN
Dropout	Attention	Dropout	Attention	CNN	CNN	MaxPooling
Dense	Dropout	Dense	Dropout	Dropout	MaxPooling	CNN
	Dense		Dense	MaxPooling	Flatten	MaxPooling
				Flatten	BiLSTM/LSTM	Flatten
				Dense	Dense	BiLSTM/LSTM
				Dense		Attention
						Dropout
						Dense

2 METHODOLOGY

2.1 Applied Models

Table 1 shows the number, type, and order of layers of proposed models that are applied to fulfill the emotion intensity detection task. The parameter settings are as follows: convolution layers are all 1D and have 64 filters and kernel size of three. ReLU activation function is applied for adding non-linearity. Dropout layers are used as regularizers and their ratio is set to 0.1. 1D max-pooling layers, with a kernel size of four are used to introduce sparsity in the network parameters and to learn deep feature representations. Dense layers are used with the activation functions of sigmoid for finding the predicted binary distribution of the target class. The number of epochs is selected as 250 and the batch size is set to 128. The number of units in applied LSTM and BiLSTM networks is five.

2.2 Dataset

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [20] is used to train and test the proposed models. RAVDESS contains videos which provide both facial and speech features. Each video lasts approximately three seconds and contains frontal face poses of twelve female and twelve male, all north American actors and actresses, speaking and singing two lexically-matched sentences while expressing six basic emotions plus calmness, and neutral. In this study, we only used speaking records. Further, expressed emotions in RAVDESS are categorized into two levels of normal and strong intensities, which makes it a good option for emotion intensity detection.
The dataset is partitioned, in a subject independent manner, into the train and test sets, i.e., the videos related to the first eighteen actors (nine female, nine male) are used for training, and the videos of the next six subjects (three female, three male) are used for testing. Since videos are recorded with 30 fps, 30 images are extracted per second for facial analysis. However, as each video starts from a neutral state, reaches an apex, and goes back to a neutral state, the first and last twenty obtained frames per video are discarded.
Since the expressions of neutral are not categorized as normal or strong, this emotional state is not considered for emotion intensity detection. Additionally, the normal and strong expressions of calmness are only marginally different, therefore, they are omitted from the emotion intensity detection.

2.3 Features and Data Pre-processing

To obtain facial and speech features, two open-source toolkits, i.e., OpenFace [3] and openEAR [9], are used. OpenFace is able to return different features including facial landmarks, head pose, facial action units activity, and eye-gaze from both video and image inputs. The applied features in this study are facial landmarks, facial action units activity, and face rigid and non-rigid shape parameters leading to a vector of 378 elements [5]. The obtained feature values are normalized between zero and one.

[5] Other features provided by OpenFace were also investigated, however, the mentioned features resulted in the highest classification accuracy.

Table 2
The obtained accuracies for emotion intensity detection by proposed models on RAVDESS dataset based on facial and speech features (without neutral and calmness expressions). The results were obtained over ten repetitions.

(a) Facial features.

Data	LSTM		BiLSTM		CNN			CNN	
	Simple	Attention	Simple	Attention	Simple	LSTM	BiLSTM	LSTM+Attention	BiLSTM+Attention
Female and Male	53.34	52.46	53.27	54.13	55.96	55.06	54.79	55.31	**56.24**
Female	51.67	50.31	50.14	51.12	52.50	53.52	51.72	50.63	**54.72**
Male	54.24	53.72	52.81	53.66	58.38	56.45	54.26	56.97	**58.31**

(b) Speech features.

Data	LSTM		BiLSTM		CNN			CNN	
	Simple	Attention	Simple	Attention	Simple	LSTM	BiLSTM	LSTM+Attention	BiLSTM+Attention
Female and Male	63.5	61.70	67.65	69.03	69.45	68.54	69.46	60.53	**73.53**
Female	68.51	67.62	67.41	69.52	72.61	72.45	72.43	59.83	**75.67**
Male	57.36	55.30	55.97	55.21	59.73	59.18	59.72	58.87	**65.5**

openEAR is the open-source toolkit that is used for speech feature extraction. It analyses the speech signals and returns three different sets of features based on the applied configuration, i.e., INTERSPEECH 2009, emobase, and INTERSPEECH 2013. In this study, the INTERSPEECH 2009 (emo-IS09) [29] configuration is used, which leads to 384 features including minimum, maximum, and mean values of different speech features. In this study, only the MFCC and PCM set of features are used, which lead to a vector of 156 elements [6]. The obtained feature values are normalized between zero and one.

3 EXPERIMENTAL SCENARIOS AND OBTAINED RESULTS

3.1 Experiment I: Emotion Intensity Detection

As the main goal of this study is to identify the intensity of an expressed emotion by a user, the facial and speech related features of all subjects are extracted and pre-processed (as explained in Section 2.3) and are given to the proposed models (Table 1). The obtained results in Table 2 show that speech features lead to higher accuracy in emotion intensity detection than facial features. Further, Table 1 shows the combination of convolutional layer with the BiLSTM and Attention layers (CNN+BiLSTM+Att) achieves the highest performance, i.e., 73.53%, which is higher than state-of-the-art, i.e., 70.4% [12] (Table 4).
Although speech features lead to higher accuracy than facial features, Table 2.a shows that the accuracy of the models in identifying the intensity of the expressed emotions by males is higher than expressed emotions by females when facial features are used, i.e., 58.31% vs 54.72%. In comparison, when speech features are used, the obtained accuracy for females is higher than for males, i.e., 75.67% vs 65.5% (Table 2.b). La Mura [16] showed some of the speech features related to emotion recognition are related to the subject's gender. Thus, one explanation can be that females convey more details about the intensity of their emotions through their speech. To verify this hypothesis, the CNN+BiLSTM+Att model is separately applied to both the facial and speech features of each individual subject. The obtained results (Table 3.a) show that for males obtaining the emotion intensity by facial expressions is more accurate than for females, i.e., the minimum and maximum accuracies for males are 63.92% and 78.79%, respectively, while the corresponding values for females are 58.26% and 71.15%. On the other hand, Table 3.b shows finding emotion intensity via speech features for females is more accurate than males, i.e., the minimum and maximum accuracies for females are 71.49% and 95.83% while the corresponding values for males are 59.29% and 85.71%, respectively.

[6] Different combinations of features were used, however, the highest accuracy was obtained for the the applied feature set, thus, we did not use the other features. In addition, since openEAR is able to analyze a file, we did not trim the videos into smaller intervals, which reduced the running time remarkably.

Table 3
Accuracy of emotion intensity detection based on facial and speech data for males and females. The results obtained over ten repetitions.

(a) Facial features.

Male	Acc	STD	Female	Acc	STD
Sub#1	64.97	2.3	Sub#2	68.36	1.6
Sub#3	77.53	2.5	Sub#4	66.05	2.5
Sub#5	78.79	1.3	Sub#6	69.46	1.4
Sub#7	75.85	2.6	Sub#8	62.17	5.3
Sub#9	72.62	1.8	Sub#10	67.55	1.2
Sub#11	72.71	2.1	Sub#12	68.42	4.7
Sub#13	63.92	3.1	Sub#14	69.19	2.2
Sub#15	66.03	2.5	Sub#16	63.11	3.6
Sub#17	67.78	2.1	Sub#18	71.15	2.1
Sub#19	73.23	2.3	Sub#20	69.32	2.8
Sub#21	74.91	1.9	Sub#22	61.62	1.4
Sub#23	73.84	2.7	Sub#24	58.26	1.9

(b) Speech features.

Male	Acc	STD	Female	Acc	STD
Sub#1	82.86	8.3	Sub#2	88.46	4.2
Sub#3	84.21	3.0	Sub#4	87.85	4.8
Sub#5	59.29	4.8	Sub#6	85.57	4.1
Sub#7	80.71	5.8	Sub#8	75.49	4.9
Sub#9	85.71	6.3	Sub#10	71.49	3.3
Sub#11	70.49	7.3	Sub#12	78.57	6.7
Sub#13	70.01	11.0	Sub#14	74.29	3.6
Sub#15	60.00	7.6	Sub#16	79.23	5.1
Sub#17	85.71	6.7	Sub#18	95.83	4.3
Sub#19	74.29	6.9	Sub#20	67.17	8.8
Sub#21	67.50	7.3	Sub#22	72.5	9.6
Sub#23	83.50	9.8	Sub#24	95.38	5.3

Table 4
Comparison between the proposed model and the state-of-the-art for emotion intensity detection over RAVDESS on speech features in a subject independent manner.

Research	Architecture	Accuracy
Jalal [12]	CNN + BiLSTM + CapsuleNet	70.4%
Proposed model	CNN + BiLSTM + Attention	**73.53%**

3.2 Experiment II: Gender Detection

Since the obtained accuracies for emotion intensity detection for males and females are noticeably different, in this experiment we investigated the speech and facial features for the task of gender detection. As the results of the proposed models (Table 1) by using facial features were not promising, a new model was designed for this experiment. The new proposed model uses raw images of 200×200 pixels as input and consists of four layers, wherein each a 2D convolutional layer is followed by a max-pooling and a dropout layer. The kernel size of the convolution layers is 3×3, with the same padding size, and ReLU is used as the activation function. The max-pooling layer is 2×2 and dropout rates in different layers are set to 0.6, 0.4, 0.2, and 0.2, respectively. The batch size during the train and test is set to 32. The first eighteen subjects are used for training and the last six subjects are used for testing (subject independent and gender balance). The obtained accuracy of this model is 70.46%.
Repeating the experiment with the speech features led to higher accuracy for gender detection via the proposed models in Table 1. More specifically, CNN+BiLSTM+Att model obtained an accuracy of 89.8% for gender detection using the MFCC and PCM feature sets, which is the highest obtained accuracy in comparison with the other proposed models in Table 1. Table 5 shows the obtained confusion matrix by the proposed model for gender detection. We noticed that 20 of the female samples that are wrongly predicted as male belong to one subject.
A straightforward comparison between the proposed model and the state-of-the-art for gender detection task, using speech signals of RAVDESS dataset, is difficult. For instance, Singh et al. [31] performed gender detection in each individual emotion class assuming the emotion class is known. Bansal et al. [4] used only four expressions of RAVDESS for gender detection and obtained an accuracy of 94.12%, and Shaqra et al. [30] considered six emotions and obtained an accuracy of 98.67%, while a gender detection model should be robust to various emotions. Thus, in this study, we used all expressed emotional states in RAVDESS dataset, i.e., eight emotional states, for the task of gender detection. Table 6 compares the obtained accuracy by the proposed model with the state-of-the-art. Although the proposed model could not beat the state-of-the-art, it is more robust since it considers more emotional states.

Table 5
Confusion matrix for gender detection.

	Predicted Female	Predicted Male
Actual Female	143	25
Actual Male	9	159

Table 6
Comparison between the proposed model and the state-of-the-art for gender detection over RAVDESS on speech features in a subject independent manner.

Research	Model	Accuracy
Bansal et al. [4] (four emotional states)	SVM	94.12%
Shaqra et al. [30] (six emotional states)	MLP	98.67%
Proposed model (eight emotional states)	CNN + BiLSTM + Attention	89.8%

4 CONCLUSION

In this study, we designed different deep neural network based models for emotion intensity and gender detection using features obtained by open-source toolkits. The RAVDESS dataset was used to evaluate the proposed models because it is, to the best of our knowledge, the only dataset that categorizes emotions based on their intensity.
The obtained results showed a difference between the obtained accuracy of emotion intensity detection for females and males based on the applied feature set, i.e., using facial features led to more accurate results for males than for females, while using speech features led to higher accuracy for females' emotion intensity detection. Additionally, the results showed that the MFCC and PCM feature sets led to higher accuracy than facial features in emotion intensity detection. Further, we used the proposed models for gender detection task using facial and speech features. The obtained results showed that gender detection is also more accurate by using speech features than facial features for the RAVDESS dataset. In addition, the obtained results showed that the proposed model is comparable with the state-of-the-art while it is more robust in terms of handling more emotional states.

ACKNOWLEDGEMENTS

The work leading to these results has received funding from Flanders Make Proud (PROgramming by User Demonstration) and the Flemish Government under the program Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen.

Bibliography

[1] Ambadar, Z., Cohn, J.F., Reed, L.I.: All smiles are not created equal: Morphology and timing of smiles perceived as amused, polite, and embarrassed/nervous. Journal of nonverbal behavior **33**(1), 17–34 (2009)

[2] Bagheri, E., Bagheri, A., Esteban, P.G., Vanderborgth, B.: A novel model for emotion detection from facial muscles activity. In: Iberian Robotics conference. pp. 237–249. Springer (2019)

[3] Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: Openface 2.0: Facial behavior analysis toolkit. In: Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on. pp. 59–66. IEEE (2018)

[4] Bansal, M., Sircar, P.: Phoneme based model for gender identification and adult-child classification. In: 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS). pp. 1–7. IEEE (2019)

[5] Caridakis, G., Castellano, G., Kessous, L., Raouzaiou, A., Malatesta, L., Asteriadis, S., Karpouzis, K.: Multimodal emotion recognition from expressive faces, body gestures and speech. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. pp. 375–388. Springer (2007)

[6] Cohn, J.F., Kanade, T., Li, C.C.: Subtly different facial expression recognition and expression intensity estimation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 853–859 (1998)

[7] Cohn, J.F., Schmidt, K.: The timing of facial motion in posed and spontaneous smiles. In: Active Media Technology, pp. 57–69. World Scientific (2003)

[8] Ekman, P.: Facial action coding system (facs). A human face (2002)

[9] Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the 18th ACM international conference on Multimedia. pp. 1459–1462 (2010)

[10] Hamm, J., Kohler, C.G., Gur, R.C., Verma, R.: Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. Journal of neuroscience methods **200**(2), 237–256 (2011)

[11] Han, K., Yu, D., Tashev, I.: Speech emotion recognition using deep neural network and extreme learning machine. In: Fifteenth annual conference of the international speech communication association (2014)

[12] Jalal, M.A., Loweimi, E., Moore, R.K., Hain, T.: Learning temporal clusters using capsule routing for speech emotion recognition. In: Proc. Interspeech. vol. 2019, pp. 1701–1705 (2019)

[13] Jeni, L.A., Girard, J.M., Cohn, J.F., De La Torre, F.: Continuous au intensity estimation using localized, sparse facial feature space. In: 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG). pp. 1–7. IEEE (2013)

[14] Jin, Q., Li, C., Chen, S., Wu, H.: Speech emotion recognition with acoustic and lexical features. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 4749–4753. IEEE (2015)

[15] Kim, Y., Provost, E.M.: Emotion classification via utterance-level dynamics: A pattern-based approach to characterizing affective expressions. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 3677–3681. IEEE (2013)

[16] La Mura, M., Lamberti, P.: Human-machine interaction personalization: a review on gender and emotion recognition through speech analysis. In: 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT. pp. 319–323. IEEE (2020)

[17] Lee, J., Tashev, I.: High-level feature representation using recurrent neural network for speech emotion recognition. In: Sixteenth annual conference of the international speech communication association (2015)

[18] Lim, W., Jang, D., Lee, T.: Speech emotion recognition using convolutional and recurrent neural networks. In: 2016 Asia-Pacific signal and information processing association annual summit and conference (APSIPA). pp. 1–4. IEEE (2016)

[19] Liu, M., Li, S., Shan, S., Chen, X.: Au-aware deep networks for facial expression recognition. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG). pp. 1–6. IEEE (2013)

[20] Livingstone, S.R., Russo, F.A.: The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. PloS one **13**(5), e0196391 (2018)

[21] Mehrabian, A.: Nonverbal communication. Routledge (2017)

[22] Mirsamadi, S., Barsoum, E., Zhang, C.: Automatic speech emotion recognition using recurrent neural networks with local attention. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2227–2231. IEEE (2017)

[23] Mower, E., Mataric, M.J., Narayanan, S.: A framework for automatic human emotion classification using emotion profiles. IEEE Transactions on Audio, Speech, and Language Processing **19**(5), 1057–1070 (2010)
[24] Mustaqeem, M., Kwon, S., et al.: A cnn-assisted enhanced audio signal processing for speech emotion recognition. Sensors **20**(1), 183 (2020)
[25] Neumann, M., Vu, N.T.: Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech. arXiv preprint arXiv:1706.00612 (2017)
[26] Nwe, T.L., Foo, S.W., De Silva, L.C.: Speech emotion recognition using hidden markov models. Speech communication **41**(4), 603–623 (2003)
[27] Pitaloka, D.A., Wulandari, A., Basaruddin, T., Liliana, D.Y.: Enhancing cnn with preprocessing stage in automatic emotion recognition. Procedia computer science **116**, 523–529 (2017)
[28] Rozgić, V., Ananthakrishnan, S., Saleem, S., Kumar, R., Vembu, A.N., Prasad, R.: Emotion recognition using acoustic and lexical features. In: Thirteenth Annual Conference of the International Speech Communication Association (2012)
[29] Schuller, B., Steidl, S., Batliner, A.: The interspeech 2009 emotion challenge. In: Tenth Annual Conference of the International Speech Communication Association (2009)
[30] Shaqra, F.A., Duwairi, R., Al-Ayyoub, M.: Recognizing emotion from speech based on age and gender using hierarchical models. Procedia Computer Science **151**, 37–44 (2019)
[31] Singh, R., Puri, H., Aggarwal, N., Gupta, V.: An efficient language-independent acoustic emotion classification system. Arabian Journal for Science and Engineering **45**(4), 3111–3121 (2020)
[32] Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M.A., Schuller, B., Zafeiriou, S.: Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5200–5204. IEEE (2016)

The algorithm versus the chimps: On the minima of classifier performance metrics

Joep Burger[1][0000−0002−7298−5561] and
Quinten Meertens[2,3,4][0000−0002−3485−8895]

[1] Statistics Netherlands, Research and Development, CBS-weg 11, PO Box 4481, 6401 CZ Heerlen, the Netherlands
j.burger@cbs.nl
[2] Center for Nonlinear Dynamics in Economics and Finance, University of Amsterdam, Roetersstraat 11, PO Box 15867, 1001 NJ Amsterdam, the Netherlands
[3] Leiden Centre of Data Science, Leiden University, the Netherlands
[4] Statistics Netherlands, Business Statistics (The Hague), the Netherlands
q.a.meertens@uva.nl

Abstract. In this paper we seek the minima of performance metrics for binary classification to facilitate comparison between metrics and applications, and to assess the quality of inferential statistics made from non-probability samples. We use these minima to min-max normalize the performance metrics so that they can be interpreted as a percentage of the perfect classifier relative to the proverbial chimps at the zoo[†] guessing at random. We compare our results with the balanced metrics that have been introduced recently, which are corrected for bias due to class imbalance.

Keywords: Inferential statistics · Non-probability samples · Statistical learning · Supervised machine learning · Binary classification.

1 Introduction

Imagine you have to do a school exam. The test consists of one hundred multiple choice questions. At each question you can choose from four possible answers (A, B, C or D). In the Netherlands, students pass when they score 6 points or more on a scale from 0 to 10. Is it sufficient to answer sixty questions correctly to pass the test? Maybe not, because the proverbial chimps at the zoo that choose answers randomly will on average answer twenty five questions correctly. The teacher could correct for that minimum by grading twenty five correct answers a 0 and then scaling linearly to maintain a perfect score when no mistakes are made by the student, Then, a student answering sixty questions correctly fails the test, having a score of 4.7 on the scale from 0 to 10. The student would now have to answer at least seventy questions correctly to pass the test. From a statistical point of view, an advantage of this so-called min-max normalization,

[†] Inspired by Swedish physician Hans Rosling (1948–2017).

is that the students' grades are now comparable with the grades at a second school where the students can choose between two possible answers (A or B). The proverbial chimps at the zoo that answer randomly will then answer fifty question correctly, on average. After min-max normalization, a student at that second school will only pass the test when answering eighty or more questions correctly.

In supervised machine learning, algorithms instead of students are trained to find the right answer to a multiple choice question. The algorithm learns, for example, that in the case of a drawing the subject is a 'moon', 'rose' or 'fish'. (These are the first words Dutch children learn to read. Many algorithms are still in elementary school and it is nice to use something else than pictures of cats and cars as an example.) What final grade does an algorithm get for its answers to new drawings? For this purpose a considerable list of performance metrics has been developed (see, e.g., [7]). For some of them it is unclear what the score of the control group would be: how well would the chimps at the zoo perform?

In this paper, we will provide an answer to that question. The answer is important from a statistical point of view: as students' grades should preferably be comparable between subjects and schools, we would like the performance of algorithms to be comparable between metrics and applications. Moreover, we would like to have a statistical interpretation of the actual value of a performance metric, such that the interpretation is independent of the metric and application, similar to the min-max normalized grade of a multiple-choice test: a grade equal to 6 can always be interpreted as 60 percent between guessing and perfection. That statistical interpretation is essential when employing supervised machine learning algorithms at national statistical institutes, such as Statistics Netherlands, to produce official statistics.

Official statistics provide quantitative information about the status and development of well-defined populations such as businesses or households. One of the challenges is to produce such information at reasonable accuracy, cost and time. A burning question in official statistics is how to make inference from non-probability (NP) samples [8]. NP samples like social media messages or sensor data can offset some disadvantages of questionnaires sent to units in a probability sample, such as response burden, high costs and a considerable time lag between data collection and dissemination [1]. However, not all units in the population of interest have a positive and known probability of being included in an NP sample. This rules out design-based estimators from sampling theory.

Alternatively, the data-generating mechanism of NP samples can be deduced by modeling the relationship between features and the target variable in the NP sample and use it to predict the missing data, assuming they are missing at random. Statistical models could then be deployed, but more often machine learning algorithms are used, because they are designed for prediction or extrapolation and scale better with the number of features.

To assess the quality of the extrapolations, the quality of the predictions are assessed on test sets for which the actual value is known. A range of performance

metrics exists to this end [7]. However, as noted before, it remains unclear what the proverbial chimps at the zoo would achieve by randomly guessing the value of the target variable. A typical example is high accuracy in imbalanced datasets: if a class has a relative frequency of 95%, it is easy to obtain a seemingly impressive accuracy of 95% by always 'predicting' the most common class.

In this paper we seek the minima of performance metrics for binary classification to facilitate comparison between metrics and to assess the quality of inferential statistics made from NP samples. We use these minima to min-max normalize the performance metrics so that they can be interpreted as percentage of perfection relative to the performance of the proverbial chimps at the zoo. We compare our results with balanced metrics [6], which have been corrected for bias due to class imbalance. In our view, the paper is a methodological contribution with preliminary simulation results that encourage a more thorough experimental study in the future.

2 Imbalanced performance metrics

We assume that we have a test set of n data points, n_1 of which are labeled positive: $y_i = 1$, where y_i is the observed class of instance i. The fraction $\alpha = n_1/n$ is referred to as the base rate. The algorithm trained on a training set of $N - n$ data points predicts for all n instances the probability that instance i belongs to the positive class: $\hat{p}_i = \mathbb{P}(y_i = 1)$. By choosing a cutoff $0 \leq c \leq 1$ above which the probability $\hat{p}_i$ is assigned to the positive class, a 2×2 contingency table or confusion matrix can be constructed (Table 1). Optimizing cutoff c is discussed in Section 4.

Table 1. Confusion matrix for cutoff c. Cells highlighted in gray can be used to derive all other cells and metrics.

	Predicted Positive	Negative	$\sum$	
Actual Positive	X_c	$n_1 - X_c$	n_1	$TPR_c = \frac{X_c}{n_1}$
Negative	Y_c	$n_2 - Y_c$	$n_2 := n - n_1$	$TNR_c = \frac{n_2 - Y_c}{n_2} = 1 - \frac{Y_c}{n_2}$
$\sum$	$X_c + Y_c$	$n - X_c - Y_c$	n	
	$PPV_c = \frac{X_c}{X_c + Y_c}$	$NPV_c = \frac{n_2 - Y_c}{n - X_c - Y_c}$	$\alpha = \frac{n_1}{n}$	

From this confusion matrix the following well-known performance metrics are derived (left two columns of Table 2). Accuracy (ACC_c) is the fraction of all cases that is predicted correctly. The true positive rate (TPR_c), also known as sensitivity or recall, is the fraction of positively labeled cases that is predicted correctly and the true negative rate (TNR_c), also known as specificity, is the fraction of negatively labeled cases that is predicted correctly. The positive predictive value (PPV_c), also known as precision, is the fraction of predicted

positive cases that is actually labeled 'positive' and the negative predictive value (NPV_c) is the fraction of predicted negative cases that is actually labeled 'negative'. The receiver operating characteristic curve, or ROC curve, plots TPR_c against the complement of TNR_c for $0 \leq c \leq 1$. The area under the ROC curve (AUC) is used as a performance metric. Note that TPR_c will decrease with c whereas TNR_c will increase with c. This trade-off is captured by Youden's J index (J_c) or Peirce Skill Score, which is also the vertical distance between the ROC curve and the diagonal. Note also that PPV_c will become unstable at higher values of c, whereas NPV_c will become unstable at lower values of c, because the respective denominators decrease there. This trade-off is captured by markedness (MRK_c). The Matthews correlation coefficient (MCC_c) is the correlation between the actual and predicted binary classifications. The positive F_1 score (PF_{1c}) is the harmonic mean of TPR_c and PPV_c. Analogously, the negative F_1 score (NF_{1c}) is the harmonic mean of TNR_c and NPV_c. The harmonic mean is more sensitive to one of the values being low than the arithmetic mean.

Table 2. Imbalanced performance metrics and their expected value when randomly guessing the positive class with probability g.

Metric Q	Definition [7]	$\mathbb{E}[Q(g)]$
ACC_c	$\frac{n_2+X_c-Y_c}{n}$	$\alpha g + (1-\alpha)(1-g)$
TPR_c	$\frac{X_c}{n_1}$	g
TNR_c	$1-\frac{Y_c}{n_2}$	$1-g$
PPV_c	$\frac{X_c}{X_c+Y_c}$	$\alpha + O(\frac{1}{n^2})$
NPV_c	$\frac{n_2-Y_c}{n-X_c-Y_c}$	$1-\alpha+O(\frac{1}{n^2})$
AUC	$\int_{c=0}^{1} TPR_c dTNR_c$	$\frac{1}{2}$
J_c	$TPR_c + TNR_c - 1$	0
MRK_c	$PPV_c + NPV_c - 1$	$0 + O(\frac{1}{n^2})$
MCC_c	$\frac{n_2 X_c - n_1 Y_c}{\sqrt{n_1 n_2 (X_c+Y_c)(n-X_c-Y_c)}}$	$0 + O(\frac{1}{n^2})$
PF_{1c}	$\frac{2}{\frac{1}{TPR_c}+\frac{1}{PPV_c}}$	
	$= \frac{2X_c}{n_1+X_c+Y_c}$	$2\alpha g\left(\frac{1}{\alpha+g} - \frac{\alpha(1-g)}{n(\alpha+g)^3}\right) + O\left(\frac{1}{n^2}\right)$
NF_{1c}	$\frac{2}{\frac{1}{TNR_c}+\frac{1}{NPV_c}}$	
	$= \frac{2(n_2-Y_c)}{n+n_2-X_c-Yc}$	$2(1-\alpha)(1-g)\left(\frac{1}{2-\alpha-g} - \frac{(1-\alpha)g}{n(2-\alpha-g)^3}\right) + O\left(\frac{1}{n^2}\right)$

We will now formally introduce how to model the outcome of the predictions made by the proverbial chimps at the zoo. To that end, let g be the probability that a chimp at the zoo predicts the positive class. We assume that the chimps will all guess according to one and the same strategy out of the following three: they may toss a fair coin, throw a dice with n sizes, n_1 of which are labeled 'positive', or always guess the most common class (the mode), i.e.:

$$g^{\text{unif}} = \frac{1}{2}$$
$$g^{\text{prop}} = \alpha$$
$$g^{\text{mode}} = \begin{cases} 1 & \text{if } \alpha > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

Then, let X and Y be independently distributed random variables with binomial distributions $X \sim \text{Bin}(n_1, g)$ and $Y \sim \text{Bin}(n_2, g)$. Each evaluation metric is a random variable as well. Table 3 gives the expected confusion matrix. The third column of Table 2 shows how to compute the expected value of each performance metric. The proofs are provided in Appendix A.1. Five of the metrics are linear functions in the random variables X and Y, hence, it is trivial to compute their expected value. The expected value of the other six metrics take the form $\mathbb{E}[f(X, Y)]$ for a nonlinear, real-valued function f. If n is very small, these expectations could in theory be computed by the closed form expression

$$\mathbb{E}[f(X,Y)] = \sum_{l=0}^{n_1} \sum_{m=0}^{n_2} f(l,m)\mathbb{P}(X=l)\mathbb{P}(Y=m).$$

In practice, however, it might take a relatively long time to evaluate this expression if n gets large. So, unless n is small, the approximations given in Table 2 should be used. In addition, note that both X and Y have a (very small, but strictly) positive probability of being 0, in which case f might not be defined (e.g., for PPV we find $0/0$). Mathematically, the correct way to deal with this is to exclude the event by conditioning the expectations on its complement. In practice, in particular for larger values of n, the obtained value will be very close to simply skipping the terms in the summation where f is not defined.

Table 3. Expected confusion matrix when randomly guessing the positive class with probability g.

		Predicted Positive	Negative	$\sum$
Actual	Positive	$n_1 g$	$n_1(1-g)$	n_1
	Negative	$n_2 g$	$n_2(1-g)$	n_2
	$\sum$	ng	$n(1-g)$	n

3 Balanced performance metrics

Some performance metrics are biased due to class imbalance. Balanced performance metrics are obtained by rewriting the imbalanced performance metrics

as a function of the imbalance coefficient $\delta = 2\alpha - 1$ and setting δ to 0, i.e. α to $\frac{1}{2}$ [6]. Table 4 shows the balanced metrics and an approximation of their expected value when randomly guessing the positive class with probability g. The derivations of the formulas in the third column can be found in Appendix A.2.

Table 4. Balanced performance metrics and their expected value when randomly guessing the positive class with probability g.

Metric Q^b	Definition [6]	$\mathbb{E}[Q^b(g)]$
ACC_c^b	$\frac{TPR_c+TNR_c}{2}$	$\frac{1}{2}$
TPR_c^b	TPR_c	g
TNR_c^b	TNR_c	$1-g$
PPV_c^b	$\frac{TPR_c}{TPR_c-TNR_c+1}$	$\frac{1}{2} + \frac{\delta(1-g)}{2n(1+\delta)(1-\delta)g} + O\left(\frac{1}{n^2}\right)$
NPV_c^b	$\frac{TNR_c}{TNR_c-TPR_c+1}$	$\frac{1}{2} - \frac{\delta g}{2n(1+\delta)(1-\delta)(1-g)} + O\left(\frac{1}{n^2}\right)$
AUC^b	$2AUC - 1$	0
J_c^b	J_c	0
MRK_c^b	$PPV_c^b + NPV_c^b - 1$	$\frac{\delta(1-2g)}{2n(1+\delta)(1-\delta)g(1-g)} + O\left(\frac{1}{n^2}\right)$
MCC_c^b	$\frac{TPR_c+TNR_c-1}{\sqrt{(TPR_c-TNR_c+1)(TNR_c-TPR_c+1)}}$	$\frac{\delta(1-2g)}{2n(1+\delta)(1-\delta)\sqrt{g(1-g)}} + O\left(\frac{1}{n^2}\right)$
PF_{1c}^b	$\frac{2TPR_c}{TPR_c-TNR_c+2}$	$2g\left(\frac{1}{1+2g} - \frac{2\left(1-\delta(1+2g)\right)(1-g)}{n(1+\delta)(1-\delta)(1+2g)^3}\right) + O\left(\frac{1}{n^2}\right)$
NF_{1c}^b	$\frac{2TNR_c}{TNR_c-TPR_c+2}$	$2(1-g)\left(\frac{1}{3-2g} - \frac{2\left(1+\delta(3-2g)\right)g}{n(1+\delta)(1-\delta)(3-2g)^3}\right) + O\left(\frac{1}{n^2}\right)$

4 Min-max normalization and optimization

After establishing $\mathbb{E}[Q(g)]$, min-max normalization can be applied to rescale each metric so that the proverbial chimps at the zoo score 0, on average:

$$Q_c^{mmn}(g) = \frac{Q_c - \mathbb{E}[Q(g)]}{1 - \mathbb{E}[Q(g)]}. \tag{1}$$

Note that $ACC^{mmn}(g)$ equals the Heidke Skill Score [4] or Cohen's κ [3] if g is set to $\frac{X+Y}{n}$. This g is, however, not a random guessing probability. The Heidke Skill Score min-max normalizes accuracy with the expected cell frequencies, which depend on the model.

Through K-fold cross validation or bootstrapping, we obtain the quality of K classifiers trained on different partitions or bootstrap samples of the data. We propose to first average Q_{kc}^{mmn} per cutoff to determine the overall optimal cutoff c^*, that is:

$$c^* = \arg\max_c \overline{Q}_c^{mmn}, \tag{2}$$

in which

$$\overline{Q}_c^{mmn} = \frac{1}{K}\sum_{k=1}^{K} Q_{kc}^{mmn}. \tag{3}$$

Then, we propose to use the distribution of $Q_k^{mmn}(c^*)$ as a proxy for the quality of the predictions when applied to unlabeled data for making statistical inference.

5 Example: normalized F_1 scores

Figure 1 shows the F_1 performance of a fictitious binary classifier that predicts 60% of the actual positive instances correctly ($TPR = 0.6$) and 80% of the actual negative cases ($TNR = 0.8$), using g^{unif} for normalization. Similar figures for accuracy and F_1 with g^{prop} can be found in Appendix B. When the test set is balanced ($\delta = 0$, i.e. $\alpha = 0.5$), this classifier scores $PF_1 = \frac{2}{3}$ and $NF_1 = \frac{8}{11}$. The more abundant the positive class relative to the negative class, the higher the classifier scores on PF_1 and the lower on NF_1 (thin red line in left panels). One solution to this sensitivity to class imbalance is to balance the metric (thin red lines in right panels) by correcting for the bias. The alternative we propose is to min-max normalize the metric (thick red line in left panels) by relating it to the expected value when randomly guessing the positive class with probability g (thin blue lines).

Two interesting observations can be made. First, data sets with a different imbalance coefficient can be compared. The classifier performs best at $\delta \approx -0.18$ where $PF_1^{mmn} \approx 0.34$, i.e. 34% from perfection relative to tossing a fair coin. A classifier with the same TPR and TNR in an application with a higher δ scores better on PF_1 (up to $\frac{3}{4}$), the same on PF_1^b but worse on PF_1^{mmn}. Second, metrics can be compared. Before min-max normalization, the classifier scores equally well on PF_1 and NF_1 at $\delta = \frac{1}{7}$ (small white points). After min-max normalization, however, the classifier scores equally well on PF_1^{mmn} and NF_1^{mmn} at $\delta \approx 0.5$ (large white points). Between $\frac{1}{7} < \delta < 0.5$, $PF_1 > NF_1$ but $PF_1^{mmn} < NF_1^{mmn}$.

Note that $\mathbb{E}[F_1]$ is sensitive to sample size n (see Tables 2 and 4), which becomes apparent when the metric is balanced and the sample is highly imbalanced (Fig. 1, right panel, blue line).

6 Conclusion

In this paper, we propose to rescale performance metrics through min-max normalization, where the minimum is set to the expected value when randomly guessing the positive class with probability g. It should be explicitly specified which expected value the algorithm is trying to defeat. Our proposed normalization yields different results than correcting for bias due to class imbalance [6] or balancing the sample [e.g. 2; 5]. The min-max normalized metrics allow for a better comparison between applications and between metrics. Moreover, we propose to use the distribution across test sets of a normalized metric at the overall optimal cutoff as performance metric for inferential statistics. Future research could

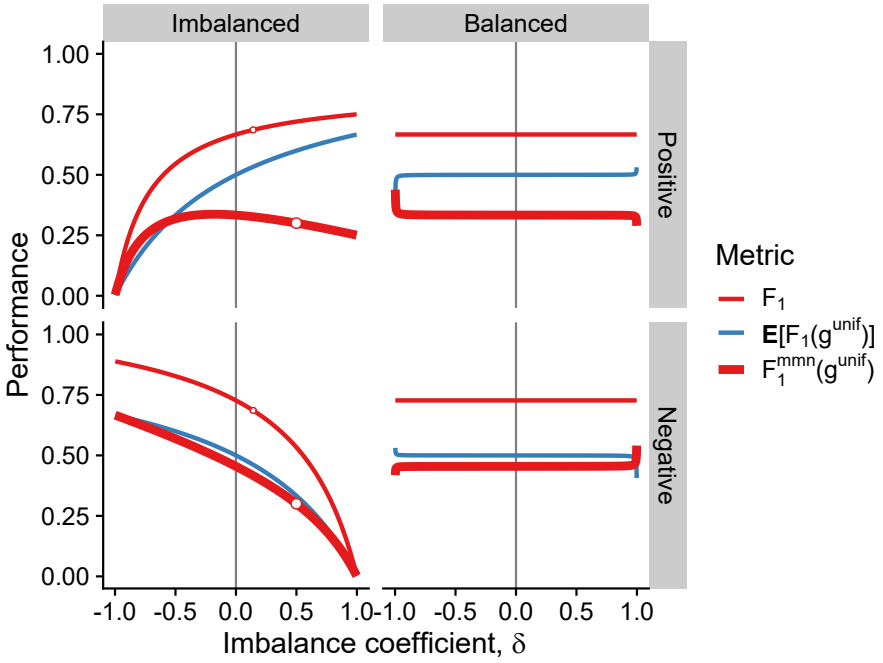


Fig. 1. Performance of a fictitious binary classifier in relation to imbalance coefficient δ. $g = 0.5$, $TPR = 0.6$, $TNR = 0.8$, $n = 1000$. White points show where positive and negative F_1 intersect.

focus on generalizing the results from binary classification to multi-class classification and regression, and on metrics that compare the predicted probability directly with the actual label, without a cutoff for constructing the confusion matrix.

Bibliography

[1] Baker, R., Brick, J.M., Bates, N.A., Battaglia, M., Couper, M.P., Dever, J.A., Gile, K.J., Tourangeau, R.: Summary report of the AAPOR task force on non-probability sampling. Journal of Survey Statistics and Methodology **1**, 90–143 (2013). https://doi.org/doi.org/10.1093/jssam/smt008

[2] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. **16**(1), 321–357 (Jun 2002). https://doi.org/10.5555/1622407.1622416

[3] Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement **20**(1), 37–46 (1960). https://doi.org/10.1177/001316446002000104

[4] Heidke, P.: Berechnung des erfolges und der gute der windstärkevorhersagen im sturmwarnungsdienst (measures of success and goodness of wind force forecasts by the gale-warning service). Geografiska Annaler **8**, 301–349 (1926). https://doi.org/10.1080/20014422.1926.11881138

[5] Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence **5**, 221–232 (2016). https://doi.org/10.1007/s13748-016-0094-0

[6] Luque, A., Carrasco, A., Martín, A., de las Heras, A.: The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition **91**, 216–231 (2019). https://doi.org/10.1016/j.patcog.2019.02.023

[7] Powers, D.M.W.: Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. Journal of Machine Learning Technologies **2**(1), 37–63 (2011)

[8] Wu, C., Thompson, M.E.: Non-probability survey samples. In: Sampling Theory and Practice, pp. 319–331. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-44246-0$_1$7

A Appendix - Proof of expected values

This appendix belongs to the paper by Burger & Meertens titled "The algorithm versus the chimps: On the minima of classifier performance metrics". It contains the proofs of the formulas provided in the third column of Table 2 and Table 4.

The key idea is to approximate an expectation of the form $\mathbb{E}[f(X,Y)]$, in which X and Y are random variables and in which f is an infinitely differentiable real-valued function, by inserting the Taylor series of f at $(\mathbb{E}[X], \mathbb{E}[Y])$. More specifically, let $x_0 = \mathbb{E}[X]$ and $y_0 = \mathbb{E}[Y]$ and consider the second-order Taylor series of f at (x_0, y_0):

$$\begin{aligned} f(x,y) \approx\ & f(x_0,y_0) + f_x(x_0,y_0)(x-x_0) + f_y(x_0,y_0)(y-y_0) \\ & + \frac{1}{2} f_{xx}(x_0,y_0)(x-x_0)^2 + \frac{1}{2} f_{yy}(x_0,y_0)(y-y_0)^2 \\ & + f_{xy}(x_0,y_0)(x-x_0)(y-y_0). \end{aligned} \tag{4}$$

We then approximate the expectation $\mathbb{E}[f(X,Y)]$ by taking the expectation of the right-hand side of the above equation. Then, assuming that X and Y are uncorrelated, we find

$$\mathbb{E}[f(X,Y)] \approx f(x_0,y_0) + \frac{1}{2} f_{xx}(x_0,y_0)\operatorname{Var}(X) + \frac{1}{2} f_{yy}(x_0,y_0)\operatorname{Var}(y). \tag{5}$$

In the proofs below we will specify the order of the approximation in terms of the size n of the test dataset.

In the remainder of the appendix, n_1 and n_2 are positive integers that sum up to n and $g \in (0,1)$ represents the probability that class 1 is predicted by the proverbial chimps at the zoo. Moreover, X will be a random variable distributed as $\mathrm{Bin}(n_1, g)$ and Y a random variable distributed as $\mathrm{Bin}(n_2, g)$. The random variables X and Y are assumed to be independent. The expectation and variance are given by

$$x_0 = \mathbb{E}[X] = n_1 g, \quad \operatorname{Var}(X) = n_1 g(1-g), \tag{6}$$

and

$$y_0 = \mathbb{E}[Y] = n_2 g, \quad \operatorname{Var}(Y) = n_2 g(1-g). \tag{7}$$

Finally, we will use the notation $\alpha = n_1/n$ (and hence $1-\alpha = n_2/n$) and $\delta = 2\alpha - 1 = (n_1 - n_2)/n$.

A.1 Expected value of imbalanced metrics

This appendix contains the derivations of the approximations of the expected values of the imbalanced performance metrics, as presented in Table 2 of the main text.

Expected Positive Predictive Value ($\mathbb{E}[PPV]$)

The positive predictive value PPV can be written as $f(X, Y)$ with $f(x, y) = x/(x + y)$. Check that

$$f_{xx}(x, y) = \frac{-2y}{(x+y)^3}, \quad f_{yy}(x_0, y_0) = \frac{2x}{(x+y)^3}. \tag{8}$$

It follows that

$$\mathbb{E}[PPV] \approx \frac{n_1 g}{ng} - \frac{n_2 g}{(ng)^3} \cdot n_1 g(1-g) + \frac{n_1 g}{(ng)^3} \cdot n_2 g(1-g) = \frac{n_1}{n} = \alpha. \tag{9}$$

The higher order terms in the Taylor series of PPV are in $O(1/n^2)$. It can be shown by looking at the terms or order 3 in the Taylor series. Only f_{xxx} and f_{yyy} remain, which are both $O(1/n^3)$ when evaluated at (x_0, y_0), and the third central moment of both X and Y are $O(n)$.

Expected Negative Predictive Value ($\mathbb{E}[NPV]$) The negative predictive value NPV can be viewed as the positive predictive value for the negative class, i.e., to compute NPV we first swap the roles of n_1 and n_2 and replace g by $1-g$ and then compute PPV. It follows that

$$\mathbb{E}[NPV] = 1 - \alpha + O\left(\frac{1}{n^2}\right). \tag{10}$$

Expected Area under the ROC curve ($\mathbb{E}[AUC]$) If the threshold value c is equal to 1, then any coin toss prediction by the chimps is considered as tails, corresponding to the point $(0, 0)$ on the ROC curve. Similarly, $c = 0$ corresponds to the point $(1, 1)$ on the ROC curve. For any other value of the threshold value c, the predictions by the chimps do not depend on c, and thus we have $TPR_c = X/n_1$ and $1 - TNR_c = Y/n_2$, for any $0 < c < 1$. The ROC curve can be obtained by connected these three points, resulting in (the random variable!)

$$\begin{aligned} AUC &= \frac{1}{2}\frac{Y}{n_2}\frac{X}{n_1} + \left(1 - \frac{Y}{n_2}\right)\frac{X}{n_1} + \frac{1}{2}\left(1 - \frac{Y}{n_2}\right)\left(1 - \frac{X}{n_1}\right) \\ &= \frac{1}{2}\left(\frac{X}{n_1} + 1 - \frac{Y}{n_2}\right). \end{aligned} \tag{11}$$

It then follows that $\mathbb{E}[AUC] = \frac{1}{2}$.

Expected Matthews Correlation Coefficient ($\mathbb{E}[MCC]$) The Matthews Correlation Coefficient (MCC) can be written as $f(X, Y)$ in which

$$f(x, y) = \frac{n_2 x - n_1 y}{\sqrt{n_1 n_2 (x+y)(n-x-y)}}. \tag{12}$$

Introducing the function $D(x,y) = n(x+y) - (x+y)^2$, the above simplifies to

$$f = (n_1 n_2)^{-\frac{1}{2}}(n_2 x - n_1 y)D^{-\frac{1}{2}}. \tag{13}$$

Both first order partial derivatives of D are equal to $n - 2(x+y)$. The identity $n_2 x_0 - n_1 y_0 = 0$ then implies that only the following term remains in $f_{xx}(x_0, y_0)$:

$$\begin{aligned} f_{xx}(x_0,y_0) &= 2 \cdot (n_1 n_2)^{-\frac{1}{2}} \cdot n_2 \cdot (-\tfrac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0,y_0) \cdot (2 - n(x_0+y_0)) \\ &= \frac{-n_2(1-2g)}{n^2\sqrt{n_1 n_2 g^3(1-g)^3}}. \end{aligned} \tag{14}$$

Notice that $f_{xx}(x_0,y_0) = O(1/n^2)$, and hence $f_{xx}(x_0,y_0)\operatorname{Var}(X) = O(1/n)$. Similarly, we obtain

$$\begin{aligned} f_{yy}(x_0,y_0) &= 2 \cdot (n_1 n_2)^{-\frac{1}{2}} \cdot (-n_1) \cdot (-\tfrac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0,y_0) \cdot (2 - n(x_0+y_0)) \\ &= \frac{n_1(1-2g)}{n^2\sqrt{n_1 n_2 g^3(1-g)^3}}. \end{aligned} \tag{15}$$

Interestingly, we have derived that

$$f_{yy}(x_0,y_0)\operatorname{Var}(Y) = -f_{xx}(x_0,y_0)\operatorname{Var}(X). \tag{16}$$

In particular, we have $f_{yy}(x_0,y_0)\operatorname{Var}(Y) = O(1/n)$. Finally, as $f(x_0,y_0) = 0$, we have shown that

$$\mathbb{E}[MCC] = 0 + O\left(\frac{1}{n^2}\right). \tag{17}$$

Expected Positive F_1 ($\mathbb{E}[PF_1]$) The positive F_1 score (PF_1) can be written as $f(X,Y)$ for $f(x,y) = 2x/(n_1 + x + y)$. Check that

$$f_{xx}(x,y) = \frac{-4(n_1+y)}{(n_1+x+y)^3}, \quad f_{yy}(x,y) = \frac{4x}{(n_1+x+y)^3}. \tag{18}$$

We leave it to the reader to check that $f_{xxx}(x_0,y_0) = O(1/n^3)$ and $f_{yyy}(x_0,y_0) = O(1/n^3)$. It follows that

$$\begin{aligned} \mathbb{E}[PF_1] &= \frac{2n_1 g}{n_1+ng} - \frac{2(n_1+n_2 g)}{(n_1+ng)^3} \cdot n_1 g(1-g) + \frac{2n_1 g}{(n_1+ng)^3} \cdot n_2 g(1-g) + O\left(\frac{1}{n^2}\right) \\ &= \frac{2n_1 g}{n_1+ng} - \frac{2n_1^2 g(1-g)}{(n_1+ng)^3} + O\left(\frac{1}{n^2}\right) \\ &= 2n_1 g\left(\frac{1}{n_1+ng} - \frac{n_1(1-g)}{(n_1+ng)^3}\right) + O\left(\frac{1}{n^2}\right) \\ &= 2\alpha g\left(\frac{1}{\alpha+g} - \frac{\alpha(1-g)}{n(\alpha+g)^3}\right) + O\left(\frac{1}{n^2}\right). \end{aligned} \tag{19}$$

Notice that $\mathbb{E}[PF_1(X,Y)] - PF_1(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n)$ and not $O(1/n^2)$. Moreover, the difference is strictly negative.

Expected Negative F_1 ($\mathbb{E}[NF_1]$) The approximation of the expectation of the negative F_1 score (NF_1) can be obtained from that of PF_1 by first swapping n_1 and n_2 and replacing g by $1-g$. In particular, we find

$$\mathbb{E}[NF_1] = 2(1-\alpha)(1-g)\left(\frac{1}{2-\alpha-g} - \frac{(1-\alpha)g}{n(2-\alpha-g)^3}\right) + O\left(\frac{1}{n^2}\right), \quad (20)$$

Again, notice that $\mathbb{E}[NF_1(X,Y)] - NF_1(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n)$ and not $O(1/n^2)$, and that the difference is strictly negative.

A.2 Expected value of balanced metrics

This appendix contains the derivations of the approximations of the expected values of the balanced performance metrics, as presented in Table 4 of the main text. The derivations are similar to those in Appendix A.1, although the outcomes are slightly different.

Expected balanced Positive Predictive Value ($\mathbb{E}[PPV^b]$) The balanced positive predictive value PPV^b can be written as $f(X,Y)$ with $f(x,y) = (x/n_1)/(x/n_1 + y/n_2)$. Check that

$$f_{xx}(x,y) = \frac{-2y/n_2}{n_1^2(x/n_1+y/n_2)^3}, \quad f_{yy}(x_0,y_0) = \frac{2x/n_1}{n_2^2(x/n_1+y/n_2)^3}. \quad (21)$$

It follows that

$$\begin{aligned}\mathbb{E}[PPV^b] &= \frac{1}{2} - \frac{n_1 g^2(1-g)}{n_1^2(2g)^3} + \frac{n_2 g^2(1-g)}{n_2^2(2g)^3} + O\left(\frac{1}{n^2}\right)\\ &= \frac{1}{2} + \frac{(n_1-n_2)(1-g)}{8n_1n_2g} + O\left(\frac{1}{n^2}\right)\\ &= \frac{1}{2} + \frac{\delta(1-g)}{2n(1+\delta)(1-\delta)g} + O\left(\frac{1}{n^2}\right). \quad (22)\end{aligned}$$

Observe that $\mathbb{E}[PPV^b(X,Y)] - PPV^b(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n)$, in contrast to $\mathbb{E}[PPV(X,Y)] - PPV(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n^2)$. However, the absolute value of the term of order $1/n$ can be bounded from above by $(1-g)/(8g)$.

Expected balanced Negative Predictive Value ($\mathbb{E}[NPV^b]$) The balanced negative predictive value NPV^b can be viewed as the balanced positive predictive value for the negative class, i.e., to compute NPV^b we first swap the roles of n_1 and n_2 and replace g by $1-g$ and then compute PPV^b. It follows that

$$\mathbb{E}[NPV^b] = \frac{1}{2} - \frac{\delta g}{2n(1+\delta)(1-\delta)(1-g)} + O\left(\frac{1}{n^2}\right). \quad (23)$$

Again, observe that $\mathbb{E}[NPV^b(X,Y)] - NPV^b(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n)$, in contrast to $\mathbb{E}[NPV(X,Y)] - NPV(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n^2)$. However, the absolute value of the term of order $1/n$ can be bounded from above by $g/(8(1-g))$.

Expected balanced Markedness ($\mathbb{E}[MRK^b]$) The expectation of the balanced markedness (MRK^b) can be approximated as follows:

$$\begin{aligned}\mathbb{E}[MRK^b] &= \mathbb{E}[PPV^b] + \mathbb{E}[NPV^b] - 1 \\ &= \frac{1}{2} + \frac{\delta(1-g)}{2n(1+\delta)(1-\delta)g} + \frac{1}{2} - \frac{\delta g}{2n(1+\delta)(1-\delta)(1-g)} - 1 + O\left(\frac{1}{n^2}\right) \\ &= \frac{\delta(1-2g)}{2n(1+\delta)(1-\delta)g(1-g)} + O\left(\frac{1}{n^2}\right). \end{aligned} \tag{24}$$

It shows that $\mathbb{E}[MRK^b(X,Y)] - MRK^b(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n)$, in contrast to $\mathbb{E}[MRK(X,Y)] - MRK(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n^2)$. However, if $g = \frac{1}{2}$, then the term of order $1/n$ is zero. If $g \neq \frac{1}{2}$, then the absolute value of the term of order $1/n$ can be bounded from above by $(1-2g)/(8g(1-g))$.

Expected balanced Matthews Correlation Coefficient ($\mathbb{E}[MCC^b]$) The balanced Matthews Correlation Coefficient (MCC^b) can be written as $f(X,Y)$ in which

$$f(x,y) = \frac{x/n_1 - y/n_2}{\sqrt{(x/n_1 + y/n_2)(2 - x/n_1 - y/n_1)}}. \tag{25}$$

Introducing the function $D(x,y) = 2(x/n_1 + y/n_2) - (x/n_1 + y/n_2)^2$, the above simplifies to

$$f = (x/n_1 - y/n_2)D^{-\frac{1}{2}}. \tag{26}$$

The first order partial derivatives of D are equal to $2/n_1 \cdot (1 - x/n_1 - y/n_2)$. The identity $x_0/n_1 - y_0/n_2 = 0$ then implies that only the following term remains in $f_{xx}(x_0,y_0)$:

$$\begin{aligned} f_{xx}(x_0,y_0) &= 2 \cdot (1/n_1) \cdot (-\tfrac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0,y_0) \cdot 2/n_1 \cdot (1 - x_0/n_1 - y_0/n_2)) \\ &= \frac{-(1-2g)}{4n_1^2\sqrt{g^3(1-g)^3}} \end{aligned} \tag{27}$$

Similarly, we obtain

$$\begin{aligned} f_{yy}(x_0,y_0) &= 2 \cdot (-1/n_2) \cdot (-\tfrac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0,y_0) \cdot 2/n_2 \cdot (1 - x_0/n_1 - y_0/n_2)) \\ &= \frac{(1-2g)}{4n_2^2\sqrt{g^3(1-g)^3}}. \end{aligned} \tag{28}$$

Finally, as $f(x_0,y_0) = 0$, it follows that

$$\begin{aligned} \mathbb{E}[MCC^b] &= \frac{-(1-2g)n_1 g(1-g)}{8n_1^2\sqrt{g^3(1-g)^3}} + \frac{(1-2g)n_2 g(1-g)}{8n_2^2\sqrt{g^3(1-g)^3}} + O\left(\frac{1}{n^2}\right) \\ &= \frac{(n_1 - n_2)(1-2g)}{8n_1 n_2\sqrt{g(1-g)}} + O\left(\frac{1}{n^2}\right) \\ &= \frac{\delta(1-2g)}{2n(1+\delta)(1-\delta)\sqrt{g(1-g)}} + O\left(\frac{1}{n^2}\right). \end{aligned} \tag{29}$$

Once again, observe that $\mathbb{E}[MCC^b(X,Y)] - MCC^b(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n)$, in contrast to $\mathbb{E}[MCC(X,Y)] - MCC(\mathbb{E}[X],\mathbb{E}[Y]) = O(1/n^2)$. However, if $g = \frac{1}{2}$, then the term of order $1/n$ is zero. If $g \neq \frac{1}{2}$, then the absolute value of the term of order $1/n$ can be bounded from above by $(1-2g)/(8\sqrt{g(1-g)})$.

Expected balanced Positive F_1 ($\mathbb{E}[PF_1^b]$) The balanced positive F_1 score (PF_1^b) can be written as $f(X,Y)$ for $f(x,y) = (2x/n_1)/(x/n_1 + y/n_2 + 1)$. Check that

$$f_{xx}(x,y) = \frac{-4(y/n_2+1)}{n_1^2(x/n_1+y/n_2+1)^3}, \quad f_{yy}(x,y) = \frac{4x/n_1}{n_2^2(x/n_1+y/n_2+1)^3}. \tag{30}$$

It follows that

$$\begin{aligned}
\mathbb{E}[PF_1^b] &= \frac{2g}{1+2g} - \frac{2n_1(1+g)g(1-g)}{n_1^2(1+2g)^3} + \frac{2n_2g^2(1-g)}{n_2^2(1+2g)^3} + O\left(\frac{1}{n^2}\right) \\
&= 2g\left(\frac{1}{1+2g} - \frac{\Big(n_2 - (n_1-n_2)g\Big)(1-g)}{n_1n_2(1+2g)^3}\right) + O\left(\frac{1}{n^2}\right) \\
&= 2g\left(\frac{1}{1+2g} - \frac{2\Big(1-\delta(1+2g)\Big)(1-g)}{n(1+\delta)(1-\delta)(1+2g)^3}\right) + O\left(\frac{1}{n^2}\right)
\end{aligned} \tag{31}$$

The term of order $1/n$ is bounded from above by $2g^2(1-g)/(2g+1)^3$, which is at most $8/243 \approx 0.033$ at $g = 2/5$. Moreover, it is bounded from below by $-2g(1-g^2)/(2g+1)^3$, which is at least $-4/243 \cdot (7\sqrt{7}-10) \approx -0.14$ at $g = (\sqrt{7}-2)/3 \approx 0.22$.

Expected balanced Negative F_1 ($\mathbb{E}[NF_1^b]$) The approximation of the expectation of the balanced negative F_1 score (NF_1^b) can be obtained from that of PF_1^b by first swapping n_1 and n_2 and replacing g by $1-g$. In particular, we find

$$\mathbb{E}[NF_1^b] = 2(1-g)\left(\frac{1}{3-2g} - \frac{2\Big(1+\delta(3-2g)\Big)g}{n(1+\delta)(1-\delta)(3-2g)^3}\right) + O\left(\frac{1}{n^2}\right), \tag{32}$$

The term of order $1/n$ is bounded from above by $2g(1-g)^2)/(2(1-g)+1)^3$, which is at most $8/243 \approx 0.044$ at $g = 3/5$, and bounded from below by $-2(1-g)(1-(1-g)^2)/(2(1-g)+1)^3$, which is at least $-4/243 \cdot (7\sqrt{7}-10) \approx -0.14$ at $g = (5-\sqrt{7})/3 \approx 0.78$.

B Appendix - Performance of fictitious binary classifier

This appendix shows how a fictitious binary classifier with $TPR = 0.6$ and $TNR = 0.8$ performs on accuracy (Figs. 2 and 3) and F_1 (Figs. 1 and 4 when

it is min-max normalized with the expected value when randomly guessing the positive class with probability $g = 0.5$ (Figs. 2 and 1) or $g = \alpha$ (Figs. 3 and 4), as a function of imbalance coefficient δ. Shown are imbalanced metrics (left panels) and balanced metrics (right panels), which have been corrected for bias due to class imbalance.

When $g^{unif} = \frac{1}{2}$ is chosen as control, $\mathbb{E}[ACC(g^{prop})] = \frac{1}{2}$ (Fig. 2, left panel, blue line). When $g^{prop} = \alpha$ is chosen as control, $\mathbb{E}[ACC(g^{prop})]$ is a quadratic function (Fig. 3, left panel, blue line; see Table 2). As a result, the classifier is outperformed ($ACC^{mmn}(g^{prop}) < 0$) by this strategy when imbalance is large (here when $\delta < \frac{-1-\sqrt{41}}{10} \approx -0.74$ or $\delta > \frac{-1+\sqrt{41}}{10} \approx 0.54$).

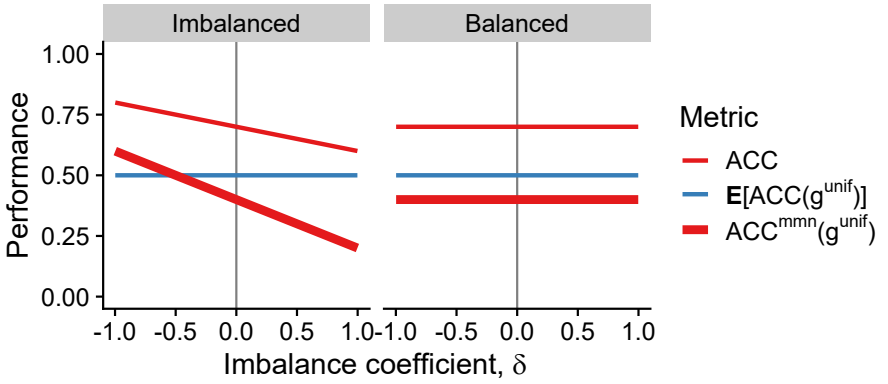


Fig. 2. Accuracy of a fictitious binary classifier in relation to imbalance coefficient δ. $g = 0.5$, $TPR = 0.6$, $TNR = 0.8$, $n = 1000$.

Before min-max normalization, the classifier scores equally well on PF_1 and NF_1 at $\delta = \frac{1}{7}$ (Fig. 4, small white points). After min-max normalization using g^{prop}, however, the classifier scores equally well on PF_1^{mmn} and NF_1^{mmn} at $\delta = -\frac{1}{3}$ (large white points). For $\delta < -\frac{1}{3}$ and $\delta > \frac{1}{7}$, the regular F_1 and normalized $F_1^{mmn}(g^{prop})$ disagree on whether the model performs better on the positive or the negative class.

By definition, the balanced F_1 is insensitive to class imbalance. After min-max normalization with g^{prop}, however, it is sensitive again to class imbalance. The larger the imbalance coefficient, the lower the classifier scores on $PF_1^{mmn,b}(g^{prop})$ and the higher on $NF_1^{mmn,b}(g^{prop})$ (Fig. 4, right panels).

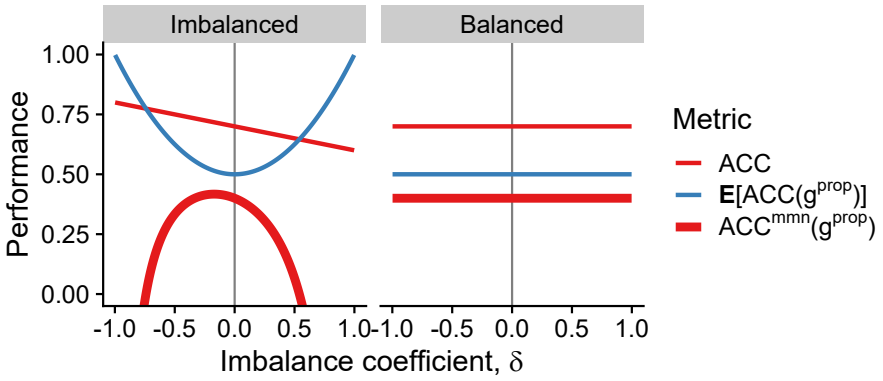


Fig. 3. Accuracy of a fictitious binary classifier in relation to imbalance coefficient δ. $g = \alpha$, $TPR = 0.6$, $TNR = 0.8$, $n = 1000$.

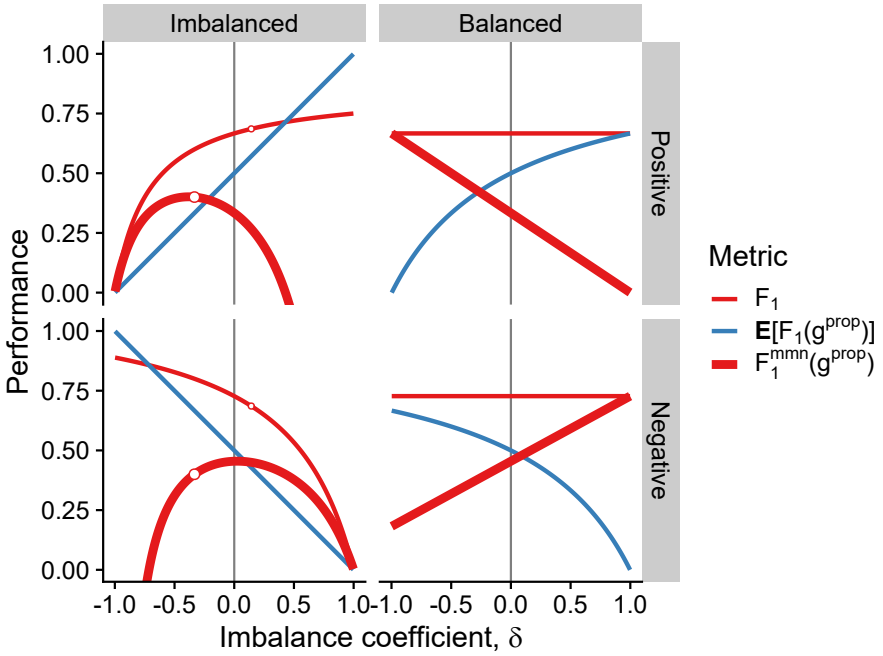


Fig. 4. F_1 of a fictitious binary classifier in relation to imbalance coefficient δ. $g = \alpha$, $TPR = 0.6$, $TNR = 0.8$, $n = 1000$. White points show where positive and negative F_1 intersect.

Philéas: Anomaly Detection for IoT Monitoring

Alberto Franzin[1][0000−0002−4066−0375]*, Raphaël Gyory[1]*, Jean-Charles Nadé[2]*, Guillaume Aubert[2], Georges Klenkle[2], and Hugues Bersini[1]

[1] IRIDIA-CoDE, Université Libre de Bruxelles, Brussels, Belgium
{afranzin,raphael.gyory,bersini}@ulb.ac.be
[2] Degetel Belgium, Brussels, Belgium
jcnade@gmail.com,gaubert@degetel.com,gklenkle@eugeka.com

Abstract. A growing number of private companies and public administrations is adopting Internet of Things (IoT) technologies to monitor resources, spaces, activities and events. Ensuring the required levels of quality of service and security is a key aspect in managing a service based on IoT. We introduce Philéas, a joint project between Degetel Belgium and the IRIDIA laboratory of the Université Libre de Bruxelles to develop a framework to analyze the activity of IoT systems and to identify possible issues via anomaly detection. In this paper we describe our framework, and we present as a demonstration two real cases that have been tackled using this framework.

Keywords: Anomaly Detection · Industrial applications · Internet of Things · Machine Learning · Quality of Service · Security.

1 Introduction

Internet of Things (IoT) devices are increasingly deployed to address a variety of real world tasks. The umbrella term Internet of Things encompasses a variety of technologies that collect, elaborate and transfer data over a network to other devices and servers in an automated and pervasive fashion [2]. They find application in a variety of domains from smart homes [42, 45] to smart cities [35, 50], from agriculture [18, 47] to manufacturing [23, 34], from healthcare [3, 29, 49] to transportation and mobility [17, 43], and many more. Due to the potential impact on society, public administrations also take a great interest in these technologies, from local to international scale [22, 37]. Though estimates on the actual number of devices vary wildly, there is strong consensus on the fact that the exponential growth will only continue in the next years, reaching the tens of billions of devices in the very near future [12, 36].

The growth of IoT technologies is however not without concerns. The complexity and scale of their applications, their ubiquity and interconnection, the heterogeneity and limited capabilities of technologies involved, the collection and treatment of personal and/or sensitive data are all factors that pose serious

*A.F., R.G. and J-C.N. contributed equally to this work.

challenges regarding energy management, security and privacy [15, 19]. IoT technologies offer several vulnerabilities for malicious actors to exploit or, simply, for faults and issues to happen [7]. The timely detection of such issues plays a key role in the management of an IoT network and application.

Big Data technologies and Artificial Intelligence (AI), in particular Machine Learning (ML), techniques are instrumental in assisting human operators in the monitoring, analysis and resolution of such issues [21, 26, 28, 31, 32, 40, 46, 52]. Devices collect a huge amount of data that is sent to other devices and central servers, where it is stored to be processed. While the data collected can be analyzed for the specific application, the metadata consisting in the message headers is useful to understand the state of the network and application. In large amounts of data, patterns of behaviour are likely to emerge, and deviations from them can be an indication of potential problems.

In this work we introduce Philéas, a framework conceived to assist managers and operators of IoT networks and applications in the analysis of IoT data. In particular, while in this project we have analyzed several different cases and applications, our main focus is the detection of anomalies in the metadata received from the devices. Philéas focuses on the analysis of anomalies from a centralized, application perspective, as independent as possible from the technical details of the devices and the network protocol, and not relying on external information about the network status. Philéas is a joint project between Degetel,[1] a consulting and services group specialized in digital transformation in France and Belgium with 15 years of experience in the IoT domain, and IRIDIA, the AI laboratory of the Université Libre de Bruxelles (ULB). Philéas answers specific market demands from the clients of Degetel regarding the management and securitization of IoT infrastructures. The project is funded by Innoviris, the institute of the Brussels Capital Region for technological innovation, with the goal of transferring academic knowledge into the industrial domain, and aims at exploring advanced AI solutions that can accompany more traditional approaches.

In the following Section we review the context of this project, the challenges in IoT systems that we address within this project and some existing relevant approaches. In Section 3 we describe the Philéas framework, including its infrastructure and the algorithms we implemented. In Section 4 we present two real world cases tackled in this project, before concluding in Section 5.

2 Background and related work

2.1 Internet of Things

Internet of Things (IoT) is a set of technologies based on uniquely identifiable devices capable of communicating with each other over a network without human interaction [2, 7]. Though the definition is rather broad, with IoT we usually refer to low power embedded devices with limited computational capabilities devoted to one task, or a specific set of tasks. A common characteristic of IoT devices

[1] https://www.degetel.com

is their pervasivity, that is, the possibility of deploying them in countless places and applications. They enable the collection of huge amounts of data, which is usually collected and analyzed. IoT devices include sensors and smart meters that measure one single value or event (e.g. temperature, humidity, the opening of a door, the failure of a mechanical component in an industrial machine) and transmit it to a central server. The use of IoT in the industry is at the base of the so-called fourth industrial revolution [23]. But in IoT we can also include vehicles capable of communicating with other vehicles and the environment (e.g. road infrastructure) [20]. IoT is also progressively entering private homes, with home automation and intelligent appliances [45].

From an economic perspective, IoT technologies create a new market, whose actors are device manufacturers, network and service providers, and application developers. Public administrations also play a role in this: for example, in Brussels public entities provide support for a smart city initiative[2] and for companies and start-ups to bring AI and IoT solutions to the market. There is also great public interest in the next generation of IoT networks, based on 5G.

Alongside with the many opportunities, the deployment of IoT solutions presents several technical challenges, from the non-interoperability of solutions, to device obsolescence, to the computational challenges of big data analysis [4]. But also the pervasivity of the devices, the possible threats to privacy and security and their implications, even at international level, are key concerns for developers, institutions and regulators [14, 16, 30].

Technical specifications of IoT devices and their interconnection encompass the full stack from the design of the electronic components of a device to the platform and application. Issues can happen at various levels of the IoT system – physical, network, application. Among the several problems that affect IoT systems, here we review the ones that concern the scope of Philéas, and we compile the following list from the management and application perspective, that is, from the central administration of the system.

2.2 Issues and challenges in IoT system administration

Device-related issues A very common situation, especially when using technologies that are inexpensive or with low capabilities, is the malfunctioning of a device, which could fail in some of its parts (e.g. measuring a wrong value, being unable to send or receive messages) or stop working altogether.

Network failure Similarly, a system can fail at the network level, e.g. because of a malfunctioning gateway. Packets can also be lost simply because of a poor network status, caused for example by bad weather conditions. In this case we typically observe a deviation from the usual patterns for a group of devices belonging to the same network, or connected to each other.

[2] https://smartcity.brussels/

Malicious action IoT systems can be the target of criminals with the goal of stealing information, or simply disrupt a service to cause financial damage. Several kinds of attacks are possible on an IoT system, for example, Distributed Denial of Service (DDoS) attacks can compromise a gateway, while the lack of end-to-end message integrity check could be exploited to alter the payload [39].

However, from the perspective of this project, the effect of malicious action results in issues that affect the system at the device or network level, or both. In fact, for both network and device failure a mere log analysis is usually insufficient to distinguish the causes of the failure, whether accidental or caused by malicious actors, and additional knowledge is required to establish the causes of the issue.

System heterogeneity The huge variety in the technologies available makes it very difficult to provide generalized solutions, even for the same kind of task. As an example, and the case that concerns Philéas the most, there are several communication protocols that can be implemented to transmit messages between devices in a network, many of which are proprietary.

2.3 Anomaly detection for IoT system management

The complexity of IoT systems is a perfect application for AI technologies and, in particular, data mining and ML techniques that can be used to process the vast amount of data and metadata collected [26, 28, 46, 52]. A complete review is beyond the scope of this project and of this work; here we limit our discussion to an overview of the techniques that have been applied to monitor the state of IoT systems from the data collected, notably anomaly detection.

An *anomaly* (or *outlier*) is an observation, or a group of observations, that exhibits two characteristics: it differs significantly from the majority of other observations, and it appears rarely in the dataset [5, 25, 27]. Anomalies can take many different forms: the simplest way to define what an anomaly is is therefore to define what *normal* observations (*inliers*) are, and to mark as anomalies all the observations that cannot be considered inliers. The nature of the deviation depends on the particular context and application. We can search for observations that deviate from the regular behaviour in the entire dataset; in this case we are considering *global* anomalies. But anomalies can also occur with respect to a subset of the data, and in this case we identify them as *local* anomalies.

Clustering and neighbourhood-based methods Clustering often is the first step taken to make sense of the data collected. By associating related observations, we can identify groups of devices that exhibit similar behaviour, for some suitable definition of similarity that takes into account relevant features. There are however many possible similarity criteria, and many clustering techniques available, each one possibly entailing different outcomes. Popular techniques include centroid-based algorithms, such as the k-means algorithms, where some observations are chosen as representatives (*centroids*) of the clusters they belong, and the remaining observations are associated to the closest centroid. Another approach is based on *density*, where a cluster is composed by points that have a

minimum amount of neighbouring points under a certain distance; in algorithms from this class, such as DBSCAN, sparse points can be considered outliers. For thorough reviews of clustering algorithms, we refer to [1, 48].

In the case of IoT log analysis, to cluster similar observations we usually need to define a distance function based on a subset of the packet fields. For example, we can group observations by the behaviour they describe, e.g. the number of packets sent by each device in a certain interval of time. But we can also analyze the aggregate of the packets sent by one device, or the devices of a specific client or a certain geographic area.

A notion of proximity between observations is also at the base of the Local Outlier Factor (LOF) algorithm, an anomaly detection technique based on the notion of *local density*, that is, how close each point is to its k neighbours [8]. In a nutshell, LOF classifies as anomalies data points whose local density differs from the local density of its neighbours. LOF is a generic technique that can be applied to various tasks for which we can define a distance between observations.

One-class learning Another approach to anomaly detection is to have a model learn only the "normal" behaviour; this corresponds to a classification task with a single target class. Anomalies are then the observations for which the model performs poorly. Techniques in this family include one-class Random Forests [24] and one-class Support Vector Machines [11]. Isolation Forests exploits the low frequency of outliers to isolate them in leaves of decision trees [33].

A family of artificial neural networks called *autoencoders* is another effective approach to anomaly detection [51]. Autoencoders perform two subsequent actions: first they map (encode) the input to a reduced space of neurons, in order to approximate the input; then they try to re-generate (decode) the input from this approximation. During the training phase they effectively learn a noise-free version of the original model, hence, in general, the reconstruction error will be smaller for observations that match the input model relatively well, rather than for observations that deviate significantly from the majority of the other points in the dataset; the first ones can therefore be considered inliers, while the latter will be identified as outliers.

Time series analysis As devices send packets to the central server either periodically or based on events, the data collected can often be modeled as multivariate time series. A multivariate time series is an ordered set of k-dimensional vectors $\mathbf{X} = \{\mathbf{x}_t\}_{t \in T}$ where each vector $\mathbf{x}_t = \{x_t^1, x_t^2, \ldots, x_t^k\}$ contains the values observed at time t. In our case, the values are the values of the different features we receive, or that we are interested to monitor in the given situation. Anomalies in time series can take different forms. We can look for a point or a sequence of points that deviates from the rest of the points in the time series (*point outliers* and *subsequence outliers*), or for a time series that exhibits a different behaviour from other time series in one or more features (*outlier time series*). Anomaly detection in time series is a rich and active field of research, thanks also to the huge importance of this task for the industry [41], and we refer to [6, 10] for detailed reviews of anomaly detection techniques for time series.

Batch vs real time analysis vs prediction Depending on the context and the specific applications, there are two possible ways of looking for anomalies in log data, batch analysis and real time analysis. Batch analysis processes data about past event, and is performed periodically or occasionally to discover anomalies occurred in the past, e.g. in the context of forensic analysis. Real time analysis is instead applied to the data as it arrives, or in small batches of recent data, and is continuously performed to ensure that potential problems are immediately spotted and taken care of. When a machine learning model is trained on the data available, it can be used to predict the future status of the IoT system, for example the reception of a packet, or the failure of a device or a network. Prediction is always applied to one single instance of the desired target.

Domain expertise Domain expertise is crucial to properly understand and evaluate the results of a data analysis, as the data alone is often not sufficient to fully understand a situation. In particular, in our applications the client needs to be involved in the process and has to analyze the outcome.

3 The Philéas framework

3.1 Scope

In Philéas we implement algorithms to detect anomalies in IoT metadata, processing logs of messages from different sources both in batch and in real time. Philéas provides a framework that can be used not only as stand-alone software, but also to develop specific solutions for different clients. Key elements in the design of the framework are the separation between the data and the AI algorithms and at the identification of common features in the various data sources. Therefore, while we can provide algorithms tailored for specific cases, in general we favour general techniques that can be applied in a variety of contexts. A specific application is then instantiated for each client, selecting the infrastructure and the algorithms that best serve the specific needs for the tasks required. The results of the anomaly detection task are meant to accompany domain expert analyses, to obtain meaningful insights about the status of the IoT system.

3.2 Network protocols

We consider two of the most common options for the communication between the devices and the central infrastructure, Sigfox [53] and LoRaWAN [13, 44]. Both are proprietary technologies, developed in France and available mostly in Europe. They operate in the ISM (Industrial, Scientific and Medical) band, at 867 – 869 MHz in Europe. The protocols have similar architectures: devices transmit packages to gateway nodes, which in turn communicate with the network server, devoted to manage the data for the various applications. The devices are not associated to a specific gateway, but rather initiate a communication by looking for an available gateway, and continue communicating with a responding one.

Sigfox is a protocol designed to be simple and robust to interference. Its packets contain only nine fields in total including the payload, with additional information about the client, and only the device ID, transmission time and RSSI of the network at the transmission as metadata usable for analysis on the receiving end. It uses an asymmetric link for transmission and reception, so it is a good choice in case of network of sensors that transmit infrequent data (e.g. temperature sensors).

LoRaWAN (Long Range Wide Area Network) is the medium access control and network layer protocol defined in the LoRa standard, designed for long range, low power connectivity: a device can function for over eight years before having to replace its battery. A LoRaWAN packet contains several metadata fields additionally to the payload. These fields include information about the gateway and the antenna, and for both uplink and downlink. LoRaWAN uses a symmetric link, so it is a better choice in case of bidirectional communication.

For more technical details about the specifications of these two protocols, we refer to their official documentations. At the moment we do not consider alternative protocols such as 5G, GPRS or NB-IoT, but the Philéas application is designed to be possibly extended to work with different packet formats.

To provide solutions as general and reusable as possible, we base our analyses on the common fields of both the Sigfox and LoRaWAN packet formats. For Sigfox, the relevant fields include, aside from the payload, the device ID, the client ID, the timestamp, and the RSSI (Received Signal Strength Indicator), a measure of the power of the radio signal, and thus on the quality of the network at the moment of the transmission. Analyzing the content of these fields can already provide several insights on the status of the IoT system. In addition to this, LoRaWAN provides several other information about the network, such as the uplink and downlink gateways, and the connection status, that can be used for deeper analyses when needed.

3.3 Algorithms

Simple rule-based and statistical analyses are very effective in several scenarios. For example, if a device did not send a message in the last x hours, or sent $y\%$ more (or less) messages than the other devices in its network, it may considered a problem. The values of x and y are to be determined by the specific application, either as fixed values provided by the client or after a preliminary data analysis. We can also compare the current behaviour of a device with its past behaviour, to observe whether it changed significantly, according to some threshold values.

In IoT networks the chances of losing a packet are comparatively high with respect to other network technologies, without this necessarily being related to actual problems. Hence, point outliers in metadata, especially multivariate (a single packet received or lost) are at high risk of being false positives. We therefore focus on subsequence outliers and outlier time series, as indications of possible persistent problems.

We use the k-means clustering to identify devices based on their transmission behaviour. For the same task in a big data context we also implement a MapRe-

duce version of the k-center clustering algorithm [9]. The distance function may depend on the specific case, protocol and application, and can therefore be defined with the user. We also use the Local Outlier Factor algorithm to identify devices that have an abnormal frequency of messages, or an abnormal amount of messages received.

While statistical and clustering methods cover most of the needs in our practical cases to characterize, respectively, individual and group behaviour of the devices. However, one of the goals of this project was to investigate more advanced machine learning models for analysis of IoT metadata, for advanced analyses of large batches of data, but also to replace manually-crafted rules and to predict future issues at the device level. We implement autoencoder neural networks, whose hyperparameters are to be set for each specific case.

3.4 Infrastructure note: exchanged place with Algorithms

The database used depends on the amount of data to be collected for each client. We have the option of using the Hadoop infrastructure for managing big data, and PostgreSQL and MongoDB as databases otherwise. Streaming data can be collected using Kafka.

The algorithms of Section 3.3 are built on top of the common Python stack of scientific libraries, based on Spark (Pyspark), Pandas and Scikit-learn for data analysis, feature augmentation and machine learning tasks. We use TensorFlow for implementing deep learning solutions, and Spark for big data analysis. Whenever possible, we use the algorithms available in the Python libraries; we however implement custom algorithms for statistical and time series analyses.

The interface for the application is built using Django and node.js. Communication with the backend is handled by REST services.

4 Use cases

Here we present two examples of issues tackled using the Philéas framework, to showcase the set of algorithms we have currently available. As they refer to specific situations of Degetel clients, the data and some of the specific details are covered by non-disclosure agreements, and we will thus omit from the following presentation any detail that may identify situations, clients or any other party involved unless specifically authorized. We can however present the computational problems, and the approaches we implemented to tackle them.

4.1 Quality of Service

The first case is about Shayp[3], a Brussels-based startup that deploys IoT water telemetry sensors to monitor water consumption in indoor locations. The sensors measure the amount of water used and transmit this value to the central server

[3] http://www.shayp.com

using Sigfox packets, with a frequency of one packet every hour. Some packets are lost, either singularly or in bursts: this normally happens due to poor network conditions. Sometimes packets from a certain device may disappear completely, in case of a faulty device or external intervention (e.g. a device is misplaced after being accidentally hit). To avoid too many false positives, we do not consider a single lost packet as an anomaly; in fact, this can situation can happen for several reasons, and it is not considered problematic in itself. However, two or more consecutive expected packets lost are considered as an anomaly to note.

The dataset for the analysis we report includes anonymized logs of 500 devices for one year of activity, each observation corresponding to one Sigfox packet received ($\sim$ 2.6M packets in total). No information available regarding users is available, and the payload is encrypted. Given the simplicity of the transmission protocol, the relevant information in each packet is only the device ID, the RSSI of the network and the timestamp of the message. The expected periodicity allows us to detect the loss of one or more packets by measuring the time elapsed between two consecutive packets received from the same device.

Monitoring the status of the network and of the devices can be done, in large part, using simple time series and statistical analyses, analyzing the time of each message, and the associated RSSI. We implement rules to detect devices with an anomalous behaviour, with respect to both the other devices in the network and the device expected behaviour.

We use this case also to describe our autoencoder approach to track lost packets. The relevant information for each message is: (i) the device ID, (ii) the RSSI value measured, and (iii) the elapsed time since the previous packet from the same device. Starting from these information, for each device we build two sequences, R_n with the n last RSSI values measured for the device (normalized in the $[0, 1]$ interval, relative to the entire dataset), and T_n, the (normalized) elapsed time between each of the last n packets received. For the autoencoder to learn the "normal" behaviour, we include in the training set only data that corresponds to packages that have at most one packet lost among its predecessors.

The input features for the autoencoder are the two sequences R_n and T_n. The autoencoder has a symmetrical architecture with an input and an output layer of $2n$ nodes, a first and last hidden layer of n nodes and a third and fourth hidden layer of $\lceil n/2 \rceil$ nodes. In our experiments we used $n = 5$, for a total of ten input features. More precisely, the network architecture is the following one:

input layer 10 nodes with ReLu activation;
first hidden layer fully connected, 5 nodes, ReLu with ℓ_1 regularization;
second hidden layer fully connected, 3 nodes, ReLu activation;
third hidden layer fully connected, 3 nodes, ReLu activation;
fourth hidden layer fully connected, 5 nodes, ReLu activation;
output layer fully connected, 10 nodes, ReLu activation.

The autoencoder then computes the reconstruction error of its input. Sequences corresponding to packets considered having normal behaviour have a lower reconstruction error than packets belonging to sequences where many previous packets have been lost. We can thus fix a threshold for the reconstruction error

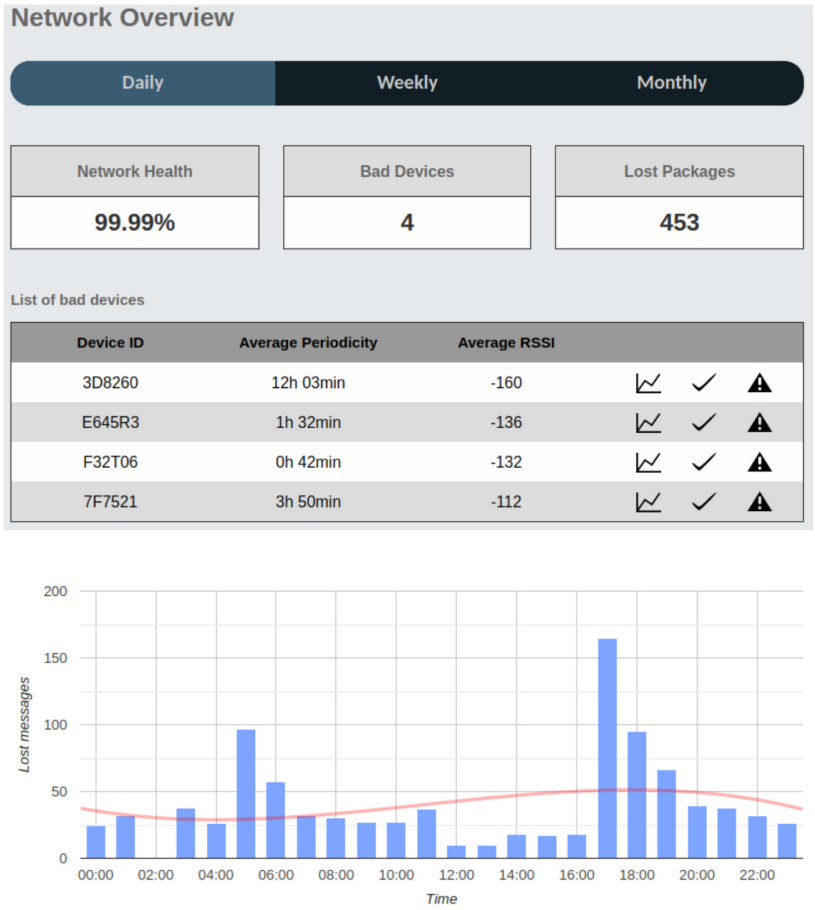


Fig. 1. Daily summary of the network status, with an overview of the main network statistics and a list of the problematic devices (top) and a plot with the hourly amount of lost packets (bottom).

to maximize the correct separation of inliers and outliers. We set the threshold experimentally on the training and validation data. This approach obtains results comparable to statistical and rule-based methods.

From the dashboard of the application the user can monitor the performance of the autoencoder, and choose to modify the error threshold, or to retrain the neural network with more recent data or different time horizons (last week, last month, etc.) should the performance decreases significantly. Philéas allows the users to filter training/validation/test data, include the features they need and set the threshold. In Figure 4.1 we show two examples of information accessible form the Philéas dashboard; the first one is a daily summary of the overall

network status, with the main statistics and a list of the devices that lost too many packets, while the second one is a plot that reports, for a given day, the hourly amount of packets expected but not received.

4.2 Distributed Denial of Service

The second case we present comes from a French company, a major actor in the local IoT market with thousand of clients throughout the entire country. They deploy LoRa sensors for a variety of tasks; with no regular frequency, these devices transmit the information they collect to the central server via LoRaWAN packets. The company requested an analysis of their logs, following an occasional service failure experienced by several of their clients on a certain day d_a, whose devices were unable to connect to the network. The company reported that, overall, nearly 45% of the connections failed, contrarily to a circa 1% of probability of connection failure under normal circumstances. The hypothesis to verify is that this was a case of DDoS [38]. Here we report the analysis on the beginning of the attack.

Due to energy considerations, devices in LoRaWAN networks are not continuously connected to the network, but rather they send a join request to the network when they have to transmit data. If the request is accepted by the server, the device will be assigned a private token to be used during the communication, that will be checked by the gateway. The connection is closed when they stop transmitting, or if they get disconnected from the network. Join requests, both accepted and rejected, are normally received and stored by the central server. Gateways are configured to cap the number of join requests they can handle in a given amount of time; when the limit is exceeded, the gateway will reject all the new incoming join requests, to preserve the central server from the additional load. The recommended practice is therefore to minimize the number of connections, and to avoid repeated retries when a join request fails or in case of a network failure, in order to minimize the load on a network. Unfortunately, IoT protocols only enforce limited secure practices by design, so it is relatively easy for a malicious actor to disrupt a service by making some devices perform an excessive amount of join requests. When this exceeds the network capacity, also non-infected devices are impacted, experiencing more join failures than usual.

We were provided six months of anonymized LoRaWAN logs, for a total of approximately two terabytes of data. The LoRaWAN packets are composed of 12 downlink fields and 60 uplink ones, only one of which is the actual payload; the other ones include many accessory information that is not necessarily useful in many contexts. Moreover, several fields have been anonymized before giving us access to the data, so only partial information was available to us. The relevant fields for this task are the device ID, the client to which the device is associated, the timestamp, and the success status. The first step is to count the packages received by each device, and by the devices of each client. We use Spark to aggregate records in the dataset by device ID, day, and client, to count daily connections. Additionally, the aggregated data is now manageable without big data algorithms or technologies.

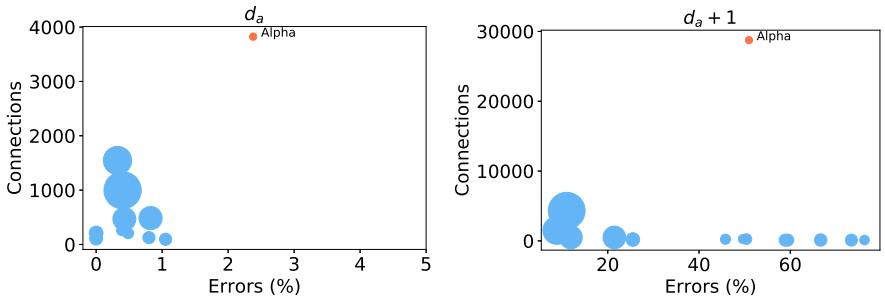


Fig. 2. Number of connections and connection error percentage for the main fifteen clients, on days d_a (the day the DDoS attack started, left plot) and $d_a + 1$ (when the attack continued, right plot). Each circle represent a client, the size of the circle is proportional to the number of devices controlled by that client. The colors indicate the outcome of the Local Outlier Factor analysis: in blue the clients that are considered inliers, in orange the client with anomalous behaviour (conventionally called Alpha).

The analysis by client shows clearly how the attack on one client impacted the other clients of the network. In Figure 4.2 we report the number of connections (on the y axis) and the error percentage (on the x axis) experienced by each client. The client that we call Alpha is the one hit by the attack and on day d_a it starts requesting an unusually high number of connections; on this day, the network load is still under control, and the other clients do not experience any particular issue. However, as the attack continues on day $d_a + 1$ and the network load increases to an excessive level, not only Alpha experiences a higher ratio of rejected joins, but also other clients become affected. In fact, all the clients experience, to various extents, an increase in the number of rejected connections. A consequence is the need for the devices to issue more join request than usual, in order to be able to transmit the information, even if, respecting the protocol recommendations, they do not issue nearly as many new join requests as the compromised client. The same outcome is observed with a Local Outlier Factor on the number of connections, normalized in the $[0, 1]$ interval, which confirms the abnormal behaviour of the devices of client Alpha.

This ex-post analysis serves also as a blueprint for the periodic monitoring of the network status. We can in fact periodically aggregate the data and spot anomalous behaviour by one or more clients; thanks to the reduced size of the aggregated dataset this can be done almost in real time.

5 Conclusions

With the growing interest in IoT technologies and applications, there is also a growing request in the market for data-based solutions to monitor IoT services. We introduced Philéas, a framework to analyze IoT logs to find anomalies in the

metadata, as an indication of potential problems in an IoT network. In Philéas we implemented several machine learning and anomaly detection techniques, and we have applied them to real-world cases of Degetel clients.

The framework can be used to implement custom solutions for clients with particular requirements; to this purpose, and depending on the requests, we are also going to include additional anomaly detection techniques, and network protocols.

Acknowledgements

The work has been made possible by the Innoviris project 2018-SHAPE-25a "PHILEAS: smart monitoring par détection de comportements anormaux appliquée aux objets connectés". M. Wattez contributed to the graphic interface and part of the implementation of the Philéas framework. We thank Shayp for the concession of using their data and use case in this paper. We thank Prof. G. Bontempi, J. De Stefani and G. Buroni for precious discussions and suggestions.

References

1. Ahmad, A., Khan, S.S.: Survey of state-of-the-art mixed data clustering algorithms. IEEE Access **7**, 31883–31902 (2019)
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer networks **54**(15), 2787–2805 (2010)
3. Baker, S.B., Xiang, W., Atkinson, I.: Internet of things for smart healthcare: Technologies, challenges, and opportunities. IEEE Access **5**, 26521–26544 (2017)
4. Banafa, A.: Three major challenges facing IoT. IEEE Internet of things (2017)
5. Ben-Gal, I.: Outlier detection. In: Data mining and knowledge discovery handbook, pp. 131–146. Springer (2005)
6. Blázquez-García, A., Conde, A., Mori, U., Lozano, J.A.: A review on outlier/anomaly detection in time series data. arXiv preprint arXiv:2002.04236 (2020)
7. Borgia, E.: The internet of things vision: Key features, applications and open issues. Computer Communications **54**, 1–31 (2014)
8. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. SIGMOD Rec. **29**(2), 93–104 (2000)
9. Ceccarello, M., Pietracaprina, A., Pucci, G.: Solving k-center clustering (with outliers) in mapreduce and streaming, almost as accurately as sequentially. arXiv preprint arXiv:1802.09205 (2018)
10. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)
11. Chen, Y., Qian, J., Saligrama, V.: A new one-class SVM for anomaly detection. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 3567–3571 (2013)
12. Cisco, C.V.N.I.: The zettabyte era—trends and analysis, 2015–2020. (2016)
13. Committee, L.A.T., et al.: Lorawan 1.1 specification. LoRa Alliance, Standard (2017)
14. Conti, M., Dehghantanha, A., Franke, K., Watson, S.: Internet of things security and forensics: Challenges and opportunities (2018)

15. Covington, M.J., Carskadden, R.: Threat implications of the internet of things. In: 2013 5th International Conference on Cyber Conflict (CYCON 2013). pp. 1–12. IEEE (2013)
16. for Cybersecurity, E.U.A.: Good practices for security of IoT. Tech. rep., European Union (02 2019)
17. Din, S., Paul, A., Hong, W.H., Seo, H.: Constrained application for mobility management using embedded devices in the internet of things based urban planning in smart cities. Sustainable Cities and Society **44**, 144 – 151 (2019)
18. Elijah, O., Rahman, T.A., Orikumhi, I., Leow, C.Y., Hindia, M.N.: An overview of internet of things (IoT) and data analytics in agriculture: Benefits and challenges. IEEE Internet of Things Journal **5**(5), 3758–3773 (2018)
19. Fu, K., Kohno, T., Lopresti, D., Mynatt, E., Nahrstedt, K., Patel, S., Richardson, D., Zorn, B.: Safety, security, and privacy threats posed by accelerating trends in the internet of things. arXiv preprint arXiv:2008.00017 (2020)
20. Gerla, M., Lee, E.K., Pau, G., Lee, U.: Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In: 2014 IEEE world forum on internet of things (WF-IoT). pp. 241–246. IEEE (2014)
21. Ghosh, A., Chakraborty, D., Law, A.: Artificial intelligence in internet of things. CAAI Transactions on Intelligence Technology **3**(4), 208–218 (2018)
22. Gil-Garcia, J.R., Pardo, T.A., Gasco-Hernandez, M.: Internet of Things and the Public Sector, pp. 3–24. Springer International Publishing (2020)
23. Gilchrist, A.: Industry 4.0: the industrial internet of things. Springer (2016)
24. Goix, N., Drougard, N., Brault, R., Chiapino, M.: One class splitting criteria for random forests. In: Zhang, M.L., Noh, Y.K. (eds.) Proceedings of the Ninth Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 77, pp. 343–358. PMLR (2017)
25. Goldstein, M., Uchida, S.: A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one **11**(4), e0152173 (2016)
26. González García, C., Núñez Valdéz, E.R., García Díaz, V., Pelayo García-Bustelo, B.C., Cueva Lovelle, J.M.: A review of artificial intelligence in the internet of things. International Journal of Interactive Multimedia and Artificial Intelligence (2019)
27. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial intelligence review **22**(2), 85–126 (2004)
28. Hussain, F., Hussain, R., Hassan, S.A., Hossain, E.: Machine learning in iot security: current solutions and future challenges. IEEE Communications Surveys & Tutorials (2020)
29. Islam, S.R., Kwak, D., Kabir, M.H., Hossain, M., Kwak, K.S.: The internet of things for health care: a comprehensive survey. IEEE access **3**, 678–708 (2015)
30. Kaska, K., Beckvard, H., Minárik, T.: Huawei, 5G and China as a security threat. NATO Cooperative Cyber Defence Center for Excellence (CCDCOE) **28** (2019)
31. Kotenko, I.V., Saenko, I., Branitskiy, A.: Applying big data processing and machine learning methods for mobile internet of things security monitoring. J. Internet Serv. Inf. Secur. **8**(3), 54–63 (2018)
32. Lee, J., Stanley, M., Spanias, A., Tepedelenlioglu, C.: Integrating machine learning in embedded sensor systems for internet-of-things applications. In: 2016 IEEE international symposium on signal processing and information technology (ISSPIT). pp. 290–294. IEEE (2016)
33. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data **6**(1) (2012)

34. Manavalan, E., Jayakrishna, K.: A review of internet of things (IoT) embedded sustainable supply chain for industry 4.0 requirements. Computers & Industrial Engineering **127**, 925–953 (2019)
35. Mehmood, Y., Ahmad, F., Yaqoob, I., Adnane, A., Imran, M., Guizani, S.: Internet-of-things-based smart cities: Recent advances and challenges. IEEE Communications Magazine **55**(9), 16–24 (2017)
36. Munirathinam, S.: Industry 4.0: Industrial internet of things (IIOT). In: Raj, P., Evangeline, P. (eds.) The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases, Advances in Computers, vol. 117, pp. 129 – 164. Elsevier (2020)
37. Ponti, M., Micheli, M., Scholten, H., Craglia, M.: Internet of things: Implications for governance (2019)
38. Salim, M.M., Rathore, S., Park, J.H.: Distributed denial of service attacks and its defenses in IoT: a survey. The Journal of Supercomputing pp. 1–44 (2019)
39. Sengupta, J., Ruj, S., Bit, S.D.: A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. Journal of Network and Computer Applications **149** (2020)
40. Sezer, O.B., Dogdu, E., Ozbayoglu, A.M.: Context-aware computing, learning, and big data in internet of things: a survey. IEEE Internet of Things Journal **5**(1), 1–27 (2017)
41. Shipmon, D.T., Gurevitch, J.M., Piselli, P.M., Edwards, S.T.: Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. arXiv preprint arXiv:1708.03665 (2017)
42. Soliman, M., Abiodun, T., Hamouda, T., Zhou, J., Lung, C.H.: Smart home: Integrating internet of things with web services and cloud computing. In: 2013 IEEE 5th international conference on cloud computing technology and science. vol. 2, pp. 317–320. IEEE (2013)
43. Solmaz, G., Wu, F., Cirillo, F., Kovacs, E., Santana, J.R., Sanchez, L., Sotres, P., Munoz, L.: Toward understanding crowd mobility in smart cities through the internet of things. IEEE Communications Magazine **57**(4), 40–46 (2019)
44. Sornin, N., Luis, M., Eirich, T., Kramp, T., Hersent, O.: Lorawan specification. LoRa alliance (2015)
45. Stojkoska, B.L.R., Trivodaliev, K.V.: A review of internet of things for smart home: Challenges and solutions. Journal of Cleaner Production **140**, 1454–1464 (2017)
46. Sun, Y., Song, H., Jara, A.J., Bie, R.: Internet of things and big data analytics for smart and connected communities. IEEE access **4**, 766–773 (2016)
47. Tzounis, A., Katsoulas, N., Bartzanas, T., Kittas, C.: Internet of things in agriculture, recent advances and future challenges. Bios. Eng. **164**, 31–48 (2017)
48. Xu, R., Wunsch, D.: Clustering, vol. 10. John Wiley & Sons (2008)
49. Yuehong, Y., Zeng, Y., Chen, X., Fan, Y.: The internet of things in healthcare: An overview. Journal of Industrial Information Integration **1**, 3–13 (2016)
50. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.: Internet of things for smart cities. IEEE Internet of Things journal **1**(1), 22–32 (2014)
51. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 665–674. ACM, New York, NY, USA (2017)
52. Zolanvari, M., Teixeira, M.A., Gupta, L., Khan, K.M., Jain, R.: Machine learning-based network vulnerability analysis of industrial internet of things. IEEE Internet of Things Journal **6**(4), 6822–6834 (2019)
53. Zuniga, J.C., Ponsard, B.: Sigfox system description. LPWAN@ IETF97, Nov. 14th **25** (2016)

Evolving Virtual Embodied Agents using External Artifact Evaluations

Lesley van Hoek, Rob Saunders, and Roy de Kleijn

Leiden Institute of Advanced Computer Science, Leiden University, Leiden, the Netherlands
{l.s.van.hoek@umail, r.saunders@liacs, kleijnrde@fsw}.leidenuniv.nl
http://www.cs.leiden.edu

Abstract. We present *neatures*, a computational art system exploring the potential of digitally evolving artificial organisms for generating aesthetically pleasing artifacts. Hexapedal agents act in a virtual environment, which they can sense and manipulate through painting. Their cognitive models are designed in accordance with theory of situated cognition. Two experimental setups are investigated: painting with a narrow- and wide perspective vision sensor. Populations of agents are optimized for the aesthetic quality of their work using a complexity-based fitness function that solely evaluates the artifact. We show that external evaluation of artifacts can evolve behaviors that produce fit artworks. Our results suggest that wide-perspective vision may be more suited for maximizing aesthetic fitness while narrow-perspective vision induces more behavioral complexity and artifact diversity. We recognize that both setups evolve distinct strategies with their own merits. We further discuss our findings and propose future directions for the current approach.

Keywords: aesthetic evaluation · artificial intelligence · artificial life · autonomous behavior · computational creativity · embodied agents · evolutionary art · neural networks · situated action · situated cognition

1 Introduction

Computational systems that produce artworks with high levels of autonomy have always provoked discussion about the definition of art and creativity. Researchers and artists working in the field of evolutionary and generative art cede control to autonomous systems that produce artworks, often intending to eliminate human intervention where possible [17]. Digital evolution is an established algorithmic process that has proven very capable of innovation [18]. In art and design, appropriate implementation of this technique can aid the generation of novel, valuable and surprising artifacts [4][2] that may be deemed creative by unbiased observers [8]. It has also been essential in the field of artificial life (a-life) [26] where researchers have been consistently surprised by creative solutions invented by artificial organisms evolving in computational environments [28]. Naturally, the process of digital evolution merely imitates life itself. The biological mechanism of natural selection is known to find and cause inventive adaptations that

enhance the survival and reproduction of organisms [15]. Consequently, these may lead to the appearance of design without a designer [10]. Adaptations may include changes in behavior. We aim our attention at a particular behavior in some non-human organisms, namely the creation of artifacts.

Several species in the natural world are known to decorate and produce structures that resemble visual art in the sense that they are intended to be attractive to potential mates. This structure creation is an important behavioral characteristic of male bower birds [12] and white-spotted pufferfishes [31]. In this paper, we explore whether artificial organisms could adapt to similar, but digitally induced pressures as a consequence of constructing artifacts. The following sections briefly discuss some challenges related to building such a computational system.

1.1 Computational aesthetic evaluation

Early examples of evolutionary art include the highly influential work of Sims [41] and Latham [47], who used genetic algorithms to mutate symbolic expressions for the composition of unpredictable yet interesting visual shapes and patterns. Both adopted a top-down approach that relies on human aesthetic judgment for the evaluation of artifacts using an interactive genetic algorithm (IGA). This technique facilitates easy exploration of large parameter spaces [44] but suffers from significant limitations: (1) IGAs rely on human evaluation at every iteration and so suffer from the *fitness bottleneck* [46], and (2) human fatigue and inconsistency make it difficult to capture universal measures [44]. Attempts to overcome these limitations have included massively multi-user systems [39] and the application of machine learning to capture user preferences [33].

Challenges in IGA helped inspire the research field of computational aesthetic evaluation (CAE), where people seek computational solutions for the assessment of human aesthetics [23]. Machado and Cardoso [29] created *NEvAr*, an autonomous system that evolves Sims' symbolic expressions with an automated evaluation procedure for images that focuses exclusively on form. Here, a speculative fitness function inspired by the study of *information aesthetics* [35] was designed which favors images that are "simultaneously, visually complex and that can be processed (by our brains) easily". In the science of aesthetics, *NEvAr*'s fitness function indicates a *formalist theory* as it proposes aesthetic experience relies on the intrinsic beauty of the artifact. In contrast, a *conceptual theory* relies on other factors that may be more important for aesthetic preference like socio-cultural contexts of the work and the previous experience of the artists and observers [40]. In a more recent publication, Redies [36] proposes a model of visual aesthetic experience that unifies these two theories. Ultimately, there is currently no agreement on which paradigm offers the most effective computational framework of human aesthetics.

1.2 Embodiment

Theorists in situated cognition view the environment as highly significant to driving human cognitive processes. Clark and Chalmers [7] suggest that the

environment directly influences an agent's behaviors as part of a two-way interaction between action and perception. Here, embodiment is key because it allows us to manipulate it to our needs. Biological brains have evolved to take advantage of the environment by offloading cognition to it through the body. Simultaneously, our visual systems evolved to rely on it more. This perspective supports the view of *externalism*, in which the cognitive process is considered something that occurs in- and outside of the mind [7]. In this context, embodiment is key to the creation of art and can be imagined as a feedback loop of action and perception occurring through a body. Brinck [5] states that the production (and consumption) of visual art can be accounted for by the theory of situated cognition [6]: "Artist and canvas form a coupled system. Artistic practice starts with gaze, and then comes the gesture that accomplishes itself when the artist is in touch with the piece [they are] working on."[5]

Experiments in the use of embodied artificial organisms and situated cognition for computational art and creativity have largely been unexplored. Thus, we present *neatures*: a prototype for an autonomous art system that simulates artificial organisms capable of producing visual art in their environment.

2 Related work

There have been several interesting art and research projects involving the use of embodied agents to create visual art. Jean Tinguely experimented with mechanical drawing machines in the 50s, exploring notions of automated artists and artificial creative processes [13]. Influences to his work can be seen in the field of swarm painting, which involves the simulation of agents supplied with some form of cognition producing emergent artworks. *Robotic Action Painter* [34] is an autonomous abstract art system based on behavioral studies of ants and other social insects. An artwork is created by employing several small wheeled robots that leave colored lines (pheromone) as they travel. A color detection sensor on each robot recognizes these lines in the environment and triggers specified behaviors for particular colors—a process analogous to *stigmergy*; a form of self-organization [14]. The result is a painting with chaotic structures that are free from preconceptions and merely represent the actions themselves. McCormack developed similar experiments using biological processes of niche construction to enhance the diversity and variation of agents' behaviors in his art system [32].

Drawing machines that take a more anthropomorphic approach can be classified as robot painters. *eDavid* [11] is an industrial robot that simulates the human painting process using a visual feedback loop to explore painterly rendering on a real canvas. Explorations in expanding its artistic skill demonstrated the possibility of expressing a given collection of images in a different style [48]. With *neatures*, we take inspiration from the flexibility of robot painters and the emerging complexity of swarms to explore the effects of aesthetic selection pressures in an evolutionary art system.

3 Implementation

Neatures[1] is a prototype computational visual art system that was developed in an attempt to employ artificial organisms for the production of visual artifacts. The current implementation is heavily inspired by the seminal work of Sims (1994), Evolving Virtual Creatures, in which a genetic algorithm was used to guide the evolution of specific abilities such as locomotion and jumping. *Neatures*' artificial organisms 'live' in three-dimensional space and are subject to physically plausible simulation. This is achieved using the Bullet physics engine [9]. The software comprises of a controller server which stores the population and commands the complete evolutionary process. A simulator client can connect to a controller and receive queries for queued rollouts. This component features a graphical user interface, allowing the user to observe the virtual organisms in real-time. The following sections briefly cover the system implementation.

3.1 Agent morphology

Virtual organisms situated in physically plausible environments are subject to strict laws of physics and, like real organisms, require an appropriate body to fulfill their purpose. Designing such a body is a difficult task, and perhaps best suited for an evolutionary process to solve. Sims [41] used a genotypic encoding of nodes and connections for the morphology of his creatures, and genetic operators, allowing for the evolution of morphology alongside control policy. In this system, a genotypic encoding scheme is used to generate a hexapod at the start of a simulation and remains fixed. The reason for this is that evolutionary optimization of morphology dramatically increases the complexity of the search landscape and is incompatible with fixed-topology neural network architectures.

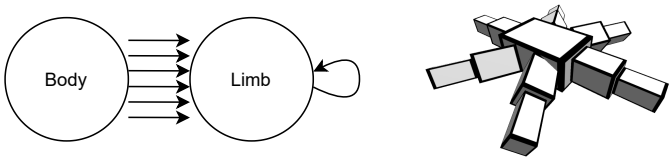


Fig. 1: Agent morphology genotype (left) and phenotype (right).

Each element stores some information about their phenotypic transformations such as size, attachment points, and node or joint type. A phenotype generation algorithm recursively traverses the graph and builds a hierarchical

[1] *Neatures* is open-source and available at https://github.com/lshoek/creative-evo-simulator

structure of boxes connected to each other by joints. Fig. 1 depicts the morphology encoding and phenotype of a hexapod. The algorithm in the current work was implemented after Krčah's example [25] with some alterations tailored to suit this work's purpose. One notable difference, for instance, is that we use a single degree of freedom per joint for simplicity.

3.2 Agent control policy

In every simulation rollout, agents are tasked to produce an artifact in their environment. In order to achieve this in *neatures*, we chose to implement a painting system. Each agent is equipped with a single brush-type node capable of applying virtual ink drops to the canvas; a specified surface area in the environment that the agent can sense and manipulate. Four invisible walls are located at a specified distance from the canvas edges to prevent agents from moving too far away from the center. Ink is only released under the conditions that the brush node is in contact with the canvas, and the agent has decided to activate it.

An agent's decision-making process and behavior are determined by its control policy. This is defined by a neural controller that continuously accepts sensory data as input, and based on this data, outputs a set of activation values. Agents sense their environment through two types of sensors: (1) a proprioceptive sensor, implemented by tracking the current joint angles and storing these in $a \in \mathbb{R}^j$, where j is equal to the number of joints in the agent's morphology and (2) a vision sensor capturing a 64x64px grayscale bitmap representation of the current canvas' content. The data of both sensors is appended to form an observation to be fed to the neural controller at regular time intervals. The physics engine and control policy are updated 60 and 20 times per second of simulated time, respectively. Fig. 2 presents the complete cognitive model of an agent.

The neural controller involves two cognitive modules; a vision model V for processing visual data inside the incoming observation, and an action model C to generate the agent's next action. V is a *convolutional variational autoencoder* (CVAE), pre-trained to compress the canvas data to a latent vector $z \in \mathbb{R}^{32}$. C

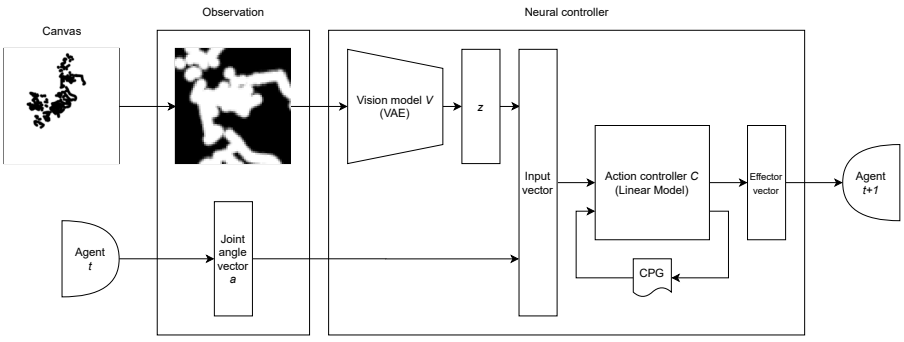


Fig. 2: Cognitive model of an agent.

is a simple linear model that takes as input a combination of latent vector z, a joint angle vector, and an additional value to stimulate continuous movement. Compression of the visual data allows the action controller to be kept small, which alleviates the credit assignment problem in difficult reinforcement learning (RL) tasks and tends to iterate faster [19]. The output layer of C uses a *tanh* activation function to output to produce a vector of effector values, including target joint angles used to update the motor parameters of the agent's joints and a value indicating the stroke width of the brush. Finally, a stimulation output value connects to a central pattern generator (CPG) after which a feedback connection to the corresponding action model input is made for the next time the neural controller is queried [24]. This minimal recurrent network structure is set up this way to evoke changing joint angle outputs. Without it, the agent would cease to move in cases where its observations remain unchanged over multiple frames and its body incidentally has zero momentum. Additionally, as sensory input drives neural excitation, it grants C control over the agent's movement speed, which could bring about more interesting behaviors. Section 4.3 describes the training procedure for V and C.

4 Experiment

We carry out two experiments where an artificial organism is evolved by optimizing for the aesthetic quality of its artifacts. The artistic medium of expression chosen for this task is painting. The main reason for this is that there exists a multitude of interesting theories and evaluation techniques of visual human aesthetics—suitable for two-dimensional content—that could be pursued to design an acceptable fitness function [16].

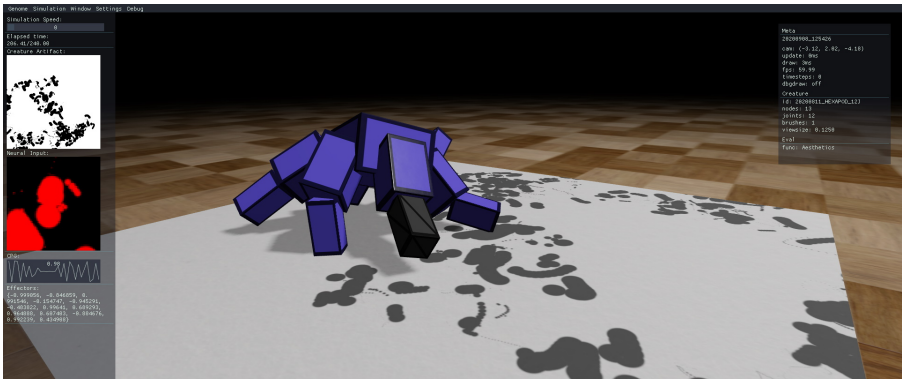


Fig. 3: The *neatures* simulator showing an agent painting.

As stated in Section 3.1, we decided to exclude morphology from evolutionary optimization, meaning we must formulate an appropriate body design for

the current experiment ourselves. We take inspiration from behavioral robotics research, where it has long been common practice to use biologically based robot designs to study artificial organisms [1]. As a matter of course, the insect-like hexapod was chosen for the current task. This design is a popular benchmark that we suppose will allow for an adequate degree of flexibility required to explore the possibilities of the virtual environment. Fig. 3 shows a screenshot of the agent as it appears in the simulator client of the system.

4.1 Setup

The following is a brief description of the realized experiments. In the first setup, the agent is supplied with a wide-perspective vision sensor. This is defined as a 64x64px grayscale bitmap representation of the environment that is equal to the size of the canvas. The orientation of this representation is at all times aligned with the facing direction of the agent and centered around the point where it last touched the canvas with its brush node. Fig. 4a shows an example of how the canvas is sensed with this perspective. The second setup supplies the agent with a narrow-perspective vision sensor, encompassing 6,5% of the canvas area as shown in Fig. 4b.

(a) **(b)**

Fig. 4: The mapping from canvas (left) to visual field (right), marked in red, for wide-perspective (4a) and narrow-perspective (4b).

The vision capabilities of the agent exist in a separate conceptual space from the one it is situated in. Agents' visual capabilities exist in *artifact space*, whereas their neural controllers output actions in *effector space*. The former is a two-dimensional representation of the environment, cultivated by the agent itself. The latter relates to objects in the three-dimensional virtual environment. Other than muscle memory (the action controller parameters), an agent has no other capabilities of memorization. As a result, the environment is the only cognitive resource to the agent by which an approximate model of situated cognition is realized. The key idea to this experiment is that, under the given conditions, a mapping between these two may be learned. If successful, the creature would be able to produce an aesthetically pleasing artifact in *artifact space* by means of its motor function in *effector space*.

4.2 Measuring aesthetic quality

After a rollout has ended, the resulting artifact is queued for fitness evaluation. In our computational environment, the fitness function is a proxy for natural selection pressures that cause the evolution of adaptations [15]. As outsiders to this virtual world, we can design this function externally, and observe what behaviors emerge from evolutionary optimization. Taking inspiration from some animal species' mate selection indicators that are attributed to external artifacts, we intentionally ignore any behavioral aspects of an agent's existence. Our fitness function is designed to evaluate images in accordance with speculative visual aesthetic theory, essentially assuming the role of an art critic.

To measure the aesthetic quality of an artifact, we use a metric closely related to Birkhoff's [3] formalist aesthetic measure, defining the formula $M = O/C$, where M is the aesthetic effectiveness, O is the degree of order and C is the degree of complexity. Birkhoff theorizes that aesthetic response to an object is stronger when the degree to which psychological effort is required to perceive it—induced by its complex features—is met with a higher degree of tension being released as the perception is realized—originating from orderly features such as symmetry and self-similarity. This formula has been disputed early and is generally regarded as inaccurate [49]. Scha & Bod [37], for instance, note that it penalizes complexity too considerably and is better suited as a measure of the degree of self-similarity. Galanter [16], however, notes that at least two aspects of Birkhoff's work remain legitimate today; the intuitive connection between aesthetic value and order/complexity relationships, and the search for a neurological base of aesthetic behavior. These aspects are reflected in the fitness function of Machado & Cardoso [29], defined in Eq. 1. Inspired by information aesthetics [35], Machado & Cardoso speculate an image's intrinsic aesthetic value to be equal to the ratio of image complexity IC to processing complexity PC.

$$reward_{aesthetic} = \frac{IC}{PC} \tag{1}$$

PC is measured at two temporal instances ($t0$ and $t1$) in the time it takes to perceive an image and provide Eq. 2. The processing complexity is maximized as PC_{t1} and PC_{t0} approach each other.

$$PC = (PC_{t0}PC_{t1})^a \left(\frac{PC_{t1} - PC_{t0}}{PC_{t1}} \right)^b \tag{2}$$

In order to find PC_{t0} and PC_{t1}, we calculate the inverse of the root mean square error (RMSE) between the original image i, and the same image after fractal compression $Fractal(i)$, as shown in Eq. 3.

$$PC_{tn} = \frac{1}{RMSE(Fractal(i),\ i)} \tag{3}$$

Machado et al. [30] compared several complexity measures with human ratings across a selected set of images in five distinct stylistic categories. Among

the results of their feature extraction experiments, their JPEG-Sobel method was found to correlate the most with human ratings, especially those related to the abstract artistic category. We calculate IC following this method as shown in Eq. 4. First, the Sobel [42] edge detection operator is applied to i horizontal and vertical directions, after which the resulting gradients are averaged. Then, JPEG compression is performed on the edges. In the dividend, size defines the total number of bytes required to store the image data.

$$IC = \frac{RMSE(Sobel(i),\ JPEG(Sobel(i)))}{size(Sobel(i))size(JPEG(Sobel(i)))} \tag{4}$$

Taylor et al. [45] note the fractal qualities of late-period action paintings by Jackson Pollock and suggests their fractal dimensions are correlated with their aesthetic qualities. Therefore, we decided to parameterize Eq. 2 using $a = 0.6$ and $b = 0.3$, increasing bias towards artifacts with more orderly features with respect to the reference implementation [29]. We argue that this suits the current experimental setup by countering excessive levels of image complexity in the artifacts due to the generally chaotic nature of agents' behaviors that generate complex and incidental painting patterns by default.

In early experiments, we found that additional encouragement to act through an easily attainable coverage reward could help agents to advance faster in early generations. This has the added benefit that a minimum specified amount of content is imposed on the artifacts. Eq. 5 defines $reward_{coverage}(x)$, where x is the mean of all normalized pixel intensities of the artifact and p is the peak coverage rate. It is essentially a smooth interpolation between x and p, ensuring a result of 1 when $x \geq p$.

$$reward_{coverage}\,(x) = 1 - sin\left(\pi \frac{\frac{1}{p}x + 1}{2}\right)^4 \tag{5}$$

with initial condition

$$x = min(x,\ p) \tag{6}$$

In our experiments, we use $p = 0.0625$, meaning that the maximum coverage reward is already reached when 6,25% of the canvas area is painted. Finally, the total artifact fitness is calculated as defined in Eq. 7. This shows the aesthetic reward is proportional to the coverage reward until peak reward p is reached, thus penalizing paintings that have little content. Table 1 presents a set of images and their fitness values.

$$fitness = 100\ reward_{coverage} + reward_{aesthetic} \times reward_{coverage} \tag{7}$$

We find these results to be satisfactory for our purposes. Although the fitness function is arguably too generous on Gaussian noise (Table 1d), such an artifact is practically impossible for an agent to produce. The Pollock-snippet (Table 1e) is evaluated far more positively and represents a more plausible result.

Table 1: A set of images and their fitness: (a) perfect symmetry, (b) an early-generation artifact with little variability in stroke width, (c) an early-generation artifact with high variability in stroke width, (d) Gaussian noise, (e) a contrast-enhanced snippet of No. 26A: Black and White by Jackson Pollock (1948).

	(a)	(b)	(c)	(d)	(e)
Fitness	101.2	116.2	145.9	399.5	871.5
Coverage	27.7%	15.5%	12.4%	18.2%	46.0%
IC	0.0697	0.4042	0.8046	44.269	48.443
PC	0.0561	0.0250	0.0175	0.0148	0.0063

4.3 Training procedure

Before any control policies can be evolved, visual model V must be pre-trained to discern between visual observations. First, 20,000 artifact samples (256x256 grayscale bitmaps) were collected in a preliminary run using an untrained visual model V. Then, a new dataset was generated by applying random affine transformations to each collected sample. This new dataset is more representative of an agent's visual observations. Finally, using the updated dataset, V was trained to encode visual observations into latent vector $z \in \mathbb{R}^{32}$ for 200 epochs.

Agents' control policies are optimized through evaluation of the quality of their work, rather than the means by which it was achieved. This indirect correspondence between goal and action may reduce credit assignment accuracy of gradient-based numerical optimization algorithms as adaptations to action controller C could have unanticipated effects on an artifact's fitness. Therefore, gradient-free methods such as evolution strategies [38] might be best suited for solving this problem. *Neuroevolution* methods have a long history of success with evolutionary robotics and have recently increased in popularity as they have been found to perform considerably well on deep RL tasks [43]. With this in consideration, we chose *covariance matrix adaptation evolution strategy* (CMA-ES) [20] for the optimization of C's parameters. Evidence shows that the algorithm performs relatively well on deceptive landscapes or sparse-reward functions up to a couple of thousands of parameters [22]. We use an open-source Python implementation of the algorithm by Hansen [21].

At the start of every evolution process, the weights of every action controller C in the population are randomly initialized with $\mu = 0$ and $\sigma = 0.1$. A population size of 32 is used, where each candidate's behavior is determined by their corresponding C, comprising 658 trainable parameters each. Every generation, one rollout is performed per agent and results in 32 artifacts. A rollout is defined as 240 seconds of simulated time an agent spends in the environment. Evaluations occur immediately after each rollout in a separate process. After all rollouts and evaluations are finished, CMA-ES uses the collected fitness values to update

each candidate's action controller parameters for the next generation. Both experiments are performed using an evolutionary process of 350 generations.

Our training setup marks several notable limitations. Foremost, the experiments are carried out separately on two mid-range laptops (i7-7700HQ/GTX1050 and i7-8750H/GTX1070), each running a single simulator client and controller server at the same time. The most significant bottleneck comes from the fractal compression procedure required for each artifact evaluation. In the current setup, we simulate two populations of 32 candidates for 350 generations and takes about 40 hours to complete. More reliable results could be collected by increasing the population size and averaging fitness over multiple rollouts for a more representative metric of the agent's general painting strategy. This is however outside of the scope of this research.

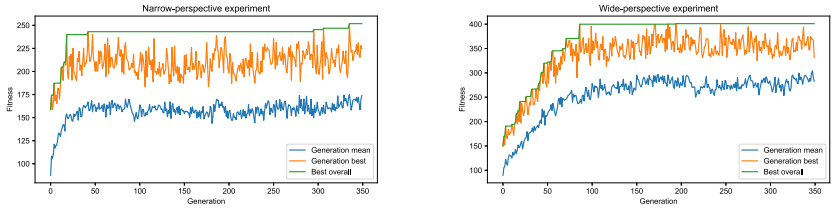


Fig. 5: Fitnesses of the narrow- (left) and wide-perspective populations (right).

5 Results

Fig. 5 presents the fitness results of the narrow- and wide-perspective vision experiments. Here, we see that the narrow-perspective population's mean fitness starts with a steep positive trend and converges towards a local optimum before the 50th generation. The wide-perspective population's mean fitness improves gradually up to around the 100th generation before a local optimum is reached. We also see that the wide-perspective population is generally about 150 points ahead of the narrow-perspective population. From these results, it is evident that the wide-perspective population performs better in terms of fitness. However, it barely shows any signs of improvement after a local optimum has been reached, until the final generation of the simulation. This is unlike the narrow-perspective population, which shows a slight upward trend around the 300th generation, and some new best-ever artifacts of the population. Table 2 presents the highest-rated artifacts of both experiments along with some key statistics. Almost every artifact shows a clear trajectory on the canvas that is telling of the strategy that was used to produce it. Fig. 6 below shows the highest-rated artifacts of the first 64 generations of both populations. We see that the sort of artifacts produced by both populations can easily be distinguished from approximately the 40th generation. From there on, we see that nearly all artifacts of the wide-vision population

Table 2: The highest-rated artifacts of all generations of wide-perspective and narrow-perspective populations and their key statistics.

Pers.	*Fitness*	*Cov.*	*IC*	*PC*	*Gen.*
Wide	401.05	36.05%	2.8339	0.0094	194
Narrow	251.82	21.33%	1.7714	0.0116	335

Wide Narrow

indicate a circular movement strategy, with little diversity among paintings. The fitness results and artifacts of this population show that this strategy is further exploited in subsequent generations, likely because of its effective contribution to maximizing fitness. In contrast, the narrow-perspective population struggles to escape a local optimum early on but demonstrates far more diversity among its artifacts in all generations. This suggests that potentially fit strategies are being explored rather than being exploited.

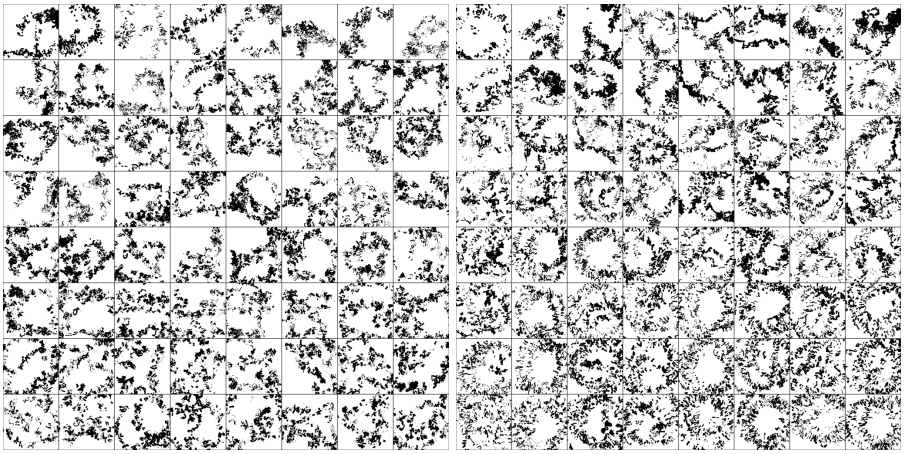


Fig. 6: The best artifacts of the first 64 generations (top-left to bottom-right) of the narrow- (left) and the wide-perspective population (right).

The discrepancy between the fitness results and the type of artifacts produced by both populations led us to believe that coverage and fitness may be strongly positively correlated. To investigate, we plotted coverage against fitness (Fig. 7) and observed that coverage is an accurate predictor of fitness in the wide-perspective population, but not necessarily for the narrow-perspective population.

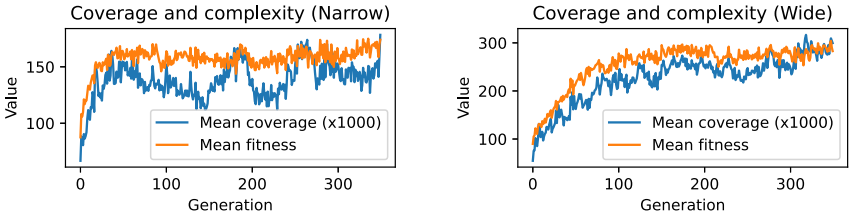


Fig. 7: Mean coverage and fitness in narrow- (left) and wide-perspective populations (right).

6 Discussion

Our results of the current experiment demonstrate a notable distinction between the narrow- and wide-perspective setup. In our experiment, we observe that virtual organisms with narrow-perspective vision trigger explorative search of the fitness landscape by the evolutionary algorithm and demonstrate more complex and distinct behavior. We also see that this is not necessarily in the interest of maximizing fitness. One explanation for this could be that relatively small adaptations to a narrow-perspective controller's weights lead to greater variations in the emerging painting strategy. In the agent's cognitive model, perception and action are closely coupled together. Therefore, distinct actions may be more likely to be triggered when visual observations are more volatile, as is the case with the narrow-perspective agents. This is in line with Brinck's [5] argument that art creation is a situated activity, noting that what the artist perceives is directly transformed into action. We further observe that narrow-perspective agents generally appear more sensitive to the environment in their painting strategies than wide-perspective agents. Narrow-perspective agents show more effective corrective behavior such as turning near the edge of the canvas. This is not as apparent in wide-perspective agents who barely appear to discernibly change their behavior near edges. Little response to edges is likely induced by the exploitation of circular movement patterns—evidently an effective strategy for painting highly fit artifacts. We further think that the widespread coverage of paint in the environment reinforces an agent's behavioral pattern. This may be due to the relatively poor compression quality of global features in visual observations of developed circular patterns, leading to similar encodings of z. Incidentally, this fact may have greatly contributed to finding the circular movement strategy.

From our observations, we theorize that volatile visual information, as demonstrated by the narrow-perspective experiment, considerably complicates the shape of the fitness landscape. For instance, a consistent circular movement strategy would be much more difficult to sustain over the length of a rollout, and over multiple generations, with narrow-perspective vision than with wide-perspective vision. Even more so, this automatically concerns any potential strategy. Although volatile visual information may impede the evolution of consistent action and perception, it does have creative merit in the sense that it elicits greater

behavioral complexity in agents. Hence, the narrow-perspective population has explored the greatest *artifact space*. This is demonstrated in Fig. 8 which presents two random selections of artifacts created in both populations.

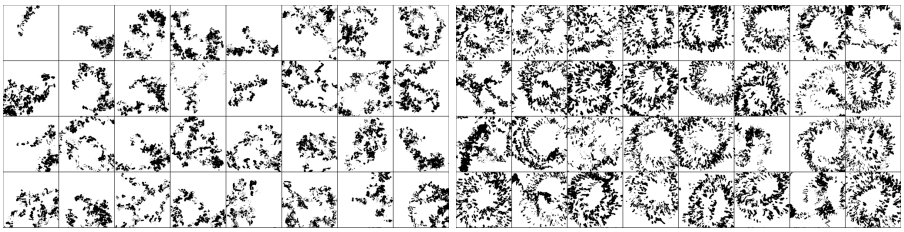


Fig. 8: Two random selections of artifacts drawn from all of the narrow-perspective population (left) and the wide-perspective population (right).

Considering our evaluation procedure; if we, hypothetically, consider Pollock's work as an aesthetic benchmark for this system (Table 1e), we consider the current fitness function helpful at guiding agents' technique towards this aesthetic up to a certain point. Fig. 7 however suggests a possible perverse instantiation problem; at least one strategy exists in which coverage can be exploited to maximize fitness. However, we believe an adjustment to the fitness function would be premature. This is because, as the fitness function is based on complexity, coverage cannot be positively correlated with fitness as it approaches 50%. The highest recorded coverage of all artifacts in both populations is 36%, whereas the coverage of our Pollock example (Table 1e) is measured at 46%. We are confident that under the current time pressure of 240 seconds, it is physically not possible for agents to cover a significantly greater part of the canvas. Therefore, we believe that agents should be assigned sufficient time so that 50% coverage could be achieved. After this is explored, we believe that a worthwhile addition to the fitness function would be a novelty reward term to overcome local optima by encouraging exploration [27].

In our experiment, we see that a proxy for selection pressures based in aesthetic properties of an external artifact can evolve a virtual organism with some success. Our agents' artificially emergent and autonomous behaviors resemble those of simple biological organisms in some ways on a superficial level and are rather interesting to observe. Whether some of the resulting artifacts are aesthetically pleasing is up to the beholder. Their chaotic patterns and compositions certainly parallel abstract expressionist action paintings to some degree. The agents' paintings share an interesting connection to this art movement as all brushstrokes represent nothing but the actions themselves. With that, one could argue for their artistic value.

6.1 Future work

We briefly propose future directions for the current research. Foremost, the system would highly benefit from a more robust visual model, as emphasized by the poor reconstruction quality of wide-perspective visual observations. This can be achieved by using a larger dataset of intermediate visual observations. Future work could assess whether granting a virtual organism continuous agency over its visual perspective, approximating the cognitive process of *attention*, is a worthwhile approach. This feature is trainable and could explore the nuance between the benefits of the demonstrated visual perspectives.

The morphology and environmental setup we chose for the task of painting is by no means the most suitable. We recommend that future work in embodied agent art should keep exploring the evolution of morphologies. This prevents authors from making predisposed choices about the most suitable body for a given task. A significant downside to this is that it requires a flexible network structure for the action controller model that is significantly more difficult to train. A search algorithm for appropriate morphology choice is another separate topic that could be further explored in the context of art-producing artificial organisms [27]. Furthermore, agents in the current work are limited to a single type of brush, paint color, and environment to explore. Therefore, future extensions could try implementing physically based painting systems, color palettes, and varying environments, each of which could bring about interesting new artifacts and behaviors. Ultimately, painting is only one method of artistic practice, and by no means the most suitable for embodied agents to practice. Computational organisms and environments allow for other artistic modes of expression to be explored such as sculpture, dance, music, poetry, etc. The possibilities are far-reaching and may one day perhaps exceed our imagination.

7 Conclusion

We have demonstrated that virtual organisms can be evolved to make aesthetically pleasing paintings using selection pressures based on aesthetic properties of the painting. The results from our experiments show notable behavioral differences between agents employed with wide-perspective and narrow-perspective vision. The wide-perspective population achieved the best results in terms of fitness by evolving a circular movement strategy effective at maximizing fitness early on, but later showing barely any signs of improvement. The narrow-perspective population performed worse and did not evolve an exploitable strategy. Instead, it brought about a diverse set of artifacts across all generations. From this we conclude that the wide-perspective setup may be more suited for maximizing aesthetic fitness while the narrow-vision setup induces more behavioral complexity and artifact diversity. Although, the scope of this research is limited, our results provided some interesting insights and discussions which provide directions for future applications of computational art systems employing virtual organisms.

References

1. Beer, R.D., Chiel, H.J., Sterling, L.S.: A biological perspective on autonomous agent design. Robotics and Autonomous Systems **6**(1-2), 169–186 (1990)
2. Bentley, P.J.: Is evolution creative. In Proc. of the AISB **99**, 28–34 (1999)
3. Birkhoff, G.D.: Aesthetic measure (p. 226). Cambridge, Mass (1933)
4. Boden, M.A.: The Creative Mind: Myths and Mechanisms. Psychology Press, Hove, UK (1990)
5. Brinck, I.: Situated cognition. On artistic creativity and aesthetic experience, dynamic systems, and art (2007)
6. Clancey, W.J.: Situated cognition: On human knowledge and computer representations. Cambridge Univ. Press (1997)
7. Clark, A., Chalmers, D.: The extended mind. Analysis **58**(1), 7–19 (1998)
8. Colton, S., Wiggins, G.A., et al.: Computational creativity: The final frontier? In: Ecai. vol. 12, pp. 21–26. Montpelier (2012)
9. Coumans, E., et al.: Bullet physics library. Open source: bulletphysics. org **15**(49), 5 (2013)
10. Dennett, D.C., Dennett, D.C.: Darwin's Dangerous Idea: Evolution and the Meanins of Life. Simon and Schuster (1996)
11. Deussen, O., Lindemeier, T., Pirk, S., Tautzenberger, M.: Feedback-guided stroke placement for a painting machine. CAe **8** (2012)
12. Diamond, J.: Animal art: variation in bower decorating style among male bowerbirds amblyornis inornatus. Proc. of the National Academy of Sciences **83**(9), 3042–3046 (1986)
13. Dohm, K., Stahlhut, H., Hoffmann, J.: Kunstmaschinen Maschinenkunst. Kehrer Verlag (2007)
14. Dorigo, M., Bonabeau, E., Theraulaz, G.: Ant algorithms and stigmergy. Future Generation Computer Systems **16**(8), 851–871 (2000)
15. Futuyma, D.J.: Natural selection and adaptation. Evolution pp. 279–301 (2009)
16. Galanter, P.: Computational aesthetic evaluation: past and future. In: Computers and creativity, pp. 255–293. Springer (2012)
17. Galanter, P.: Generative art theory. A Companion to Digital Art pp. 146–180 (2016)
18. Goldberg, D.E.: The race, the hurdle, and the sweet spot. Evolutionary design by computers pp. 105–118 (1999)
19. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)
20. Hansen, N.: The cma evolution strategy: A tutorial. arxiv (2016), preprint
21. Hansen, N., Akimoto, Y., Baudis, P.: Cma-es/pycma on github. Zenodo, doi **10** (2019)
22. Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: GECCO. pp. 1689–1696 (2010)
23. Hoenig, F.: Defining computational aesthetics. The Eurographics Assoc. (2005)
24. Hülse, M., Wischmann, S., Manoonpong, P., Von Twickel, A., Pasemann, F.: Dynamical systems in the sensorimotor loop: On the interrelation between internal and external mechanisms of evolved robot behavior. In: 50 years of artificial intelligence, pp. 186–195. Springer (2007)
25. Krčah, P.: Evolution and Learning of Virtual Robots. Ph.D. thesis, Univerzita Karlova (2016)
26. Langton, C.G.: Artificial life: An overview. Mit Press (1997)

27. Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary computation **19**(2), 189–223 (2011)
28. Lehman, J., Clune, J., Misevic, D., Adami, C., Altenberg, L., Beaulieu, J., Bentley, P.J., Bernard, S., Beslon, G., Bryson, D.M., et al.: The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. Artificial Life **26**(2), 274–306 (2020)
29. Machado, P., Cardoso, A.: All the truth about nevar. Applied Intelligence **16**(2), 101–118 (2002)
30. Machado, P., Romero, J., Nadal, M., Santos, A., Correia, J., Carballal, A.: Computerized measures of visual complexity. Acta psychologica **160**, 43–57 (2015)
31. Matsuura, K.: A new pufferfish of the genus torquigener that builds "mystery circles" on sandy bottoms in the ryukyu islands, japan (actinopterygii: Tetraodontiformes: Tetraodontidae). Ichthyological research **62**(2), 207–212 (2015)
32. McCormack, J.: Niche constructing drawing robots. In: EvoMUSART. pp. 201–216. Springer (2017)
33. McCormack, J., Lomas, A.: Understanding aesthetic evaluation using deep learning. In: EvoMUSART. pp. 118–133. Springer (2020)
34. Moura, L.: A new kind of art: The robotic action painter. X Generative Art Conf., Politecnico di Milano Univ. (2007)
35. Nake, F.: Information aesthetics: An heroic experiment. EvoMUSART **6**(2-3), 65–75 (2012)
36. Redies, C.: Combining universal beauty and cultural context in a unifying model of visual aesthetic experience. Frontiers in human neuroscience **9**, 218 (2015)
37. Scha, R., Bod, R.: Computationele esthetica. Informatie en Informatiebeleid **11**(1), 54–63 (1993)
38. Schwefel, H.P.: Numerical optimization of computer models. John Wiley & Sons, Inc (1981)
39. Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J.T., Stanley, K.O.: Picbreeder: A case study in collaborative evolutionary exploration of design space. Evolutionary computation **19**(3), 373–403 (2011)
40. Shimamura, A.P., Palmer, S.E.E.: Aesthetic science: Connecting minds, brains, and experience. OUP USA (2012)
41. Sims, K.: Artificial evolution for computer graphics. In: PACMCGIT. vol. 18, pp. 319–328 (1991)
42. Sobel, I.: An isotropic 3× 3 image gradient operator, machine vision for three-dimensional scenes (h. freeman editor) (1990)
43. Such, F.P., Madhavan, V., Conti, E., Lehman, J., Stanley, K.O., Clune, J.: Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arxiv (2017), preprint
44. Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. Proc. of the IEEE **89**(9), 1275–1296 (2001)
45. Taylor, R.P., Micolich, A.P., Jonas, D.: Fractal analysis of pollock's drip paintings. Nature **399**(6735), 422–422 (1999)
46. Todd, P.M., Werner, G.M.: Frankensteinian methods for evolutionary music. Musical networks: parallel distributed perception and performace pp. 313–340 (1999)
47. Todd, S., Latham, W.: Evolutionary art & computers. Academic Press, Inc. (1994)
48. Tresset, P., Deussen, O.: Artistically skilled embodied agents. In: AISB (2014)
49. Wilson, D.J.: An experimental investigation of birkhoff's aesthetic measure. The Journal of Abnormal and Social Psychology **34**(3), 390 (1939)

Continuous surrogate-based optimization algorithms are well-suited for expensive discrete problems

Rickard Karlsson, Laurens Bliek, Sicco Verwer, and Mathijs de Weerdt

Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Abstract. One method to solve expensive black-box optimization problems is to use a surrogate model that approximates the objective based on previous observed evaluations. The surrogate, which is cheaper to evaluate, is optimized instead to find an approximate solution to the original problem. In the case of discrete problems, recent research has revolved around surrogate models that are specifically constructed to deal with with discrete structures. A main motivation is that literature considers continuous methods, such as Bayesian optimization with Gaussian processes as the surrogate, to be sub-optimal (especially in higher dimensions) because they ignore the discrete structure by e.g. rounding off real-valued solutions to integers. However, we claim that this is not true. In fact, we present empirical evidence showing that the use of continuous surrogate models displays competitive performance on a set of high-dimensional discrete benchmark problems, including a real-life application, against state-of-the-art discrete surrogate-based methods. Our experiments on different discrete structures and time constraints also give more insight into which algorithms work well on which type of problem.

Introduction

A principal challenge in optimization is to deal with black-box objective functions. The objective function is assumed to be unknown in this case, in contrast to traditional optimization that often utilizes an explicit formulation to compute the gradient or lower bounds. Instead, we assume to have an objective $y = f(\mathbf{x}) + \epsilon$ with some unknown function $f(\mathbf{x})$ together with additive noise ϵ. Furthermore, $f(\mathbf{x})$ can be expensive to evaluate in terms of time or another resource which restricts the number of evaluations allowed.

One type of method to solve these black-box optimization problems is the use of surrogate models. Surrogate-based algorithms approximate the objective function in search of the optimal solution, with the benefit that the surrogate model is cheaper to evaluate. Bayesian optimization [22] is an example of such a surrogate-based algorithm.

An active field of research is how to deal with discrete black-box optimization problems with an expensive objective function. There are many real-world examples of this, such as deciding on the architecture of a deep neural network [7] or designing molecules with desirable properties [15]. Furthermore, optimization over structured domains was highlighted as an important problem to address from the NIPS 2017 workshop on Bayesian optimization [10].

Discrete optimization problems can be solved with a continuous surrogate model, e.g. Bayesian optimization with Gaussian processes [22], by ignoring the discrete structure and rounding off the real-valued input to discrete values. However, literature in this field generally considers this to be a sub-optimal approach [1, 8]. Therefore, research has revolved around inherently discrete models such as density estimators or decision trees, e.g. HyperOpt [2] or SMAC [12]. Another approach is to use continuous models that guarantee discrete optimal solutions, such as the piece-wise linear model IDONE [4].

In contrast to common belief, we present an empirical study that displays that continuous surrogate models, in this case Gaussian processes and linear combinations of rectified linear units, show competitive performance on expensive discrete optimization benchmarks by outperforming discrete state-of-the-art algorithms. Firstly, we will introduce the problem, the related work, and the considered benchmark problems. Then, in the remainder of the paper we 1) perform a benchmark comparison between continuous and discrete surrogate-based algorithms on optimization problems with different discrete structures (including one real-life application), 2) investigate why continuous surrogate models perform well by transforming the different discrete problem structures and visualizing the continuous surrogate models, and 3) perform a more realistic analysis that takes the time budget and evaluation time into account when comparing the algorithms. We conclude that continuous surrogates applied to discrete problems should get more attention, and leave some questions for interesting directions of future research in the domain of discrete expensive black-box optimization.

Problem Description

Consider the following class of d-dimensional discrete optimization problems:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathbb{Z}^d \\ & l_i \leq \mathbf{x}_i \leq u_i,\ i = 1, \ldots, d \end{aligned} \tag{1}$$

where l_i and u_i are the lower and upper bound for each integer-valued decision variable x_i. For black-box optimization problems, we assume to have no closed form expression for $f : \mathbb{Z}^d \to \mathbb{R}$. The only information which can be gathered about f comes from observing the output when evaluating $f(\mathbf{x})$ given some input $\mathbf{x}$. However, in many real-world applications we will also have to deal with some noise $\epsilon \in \mathbb{R}$ such that we are given the output $y = f(\mathbf{x}) + \epsilon$. Obtaining an evaluation is also assumed to be expensive: it could require large computational power, human interaction with the system or time consuming simulations. Therefore it is of interest to obtain a solution within a limited amount of evaluations B, also known as the budget.

One way of solving this class of problems is to make use of a so called surrogate model. A surrogate model is an auxiliary function M that approximates the objective function based on the points evaluated so far. This model is cheaper to evaluate in comparison to the original black-box objective function. Given a number m of already evaluated points, the surrogate model is constructed using the evaluation history

$H = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}$. The surrogate can be utilized to predict promising points to evaluate next on. The next feasible solution $\mathbf{x}^{(m+1)}$ to evaluate on can be chosen based on this prediction. These steps, which are also described in Algorithm 1, are repeated until the budget B is reached.

Typically, an acquisition function $A(M, \mathbf{x})$ is used to propose the next point $\mathbf{x}^{(m+1)}$ to evaluate with the objective function. It predicts how promising a new point $\mathbf{x}$ is, based on a trade-off of exploitation (searching at or near already evaluated points that had a low objective) and exploration (searching in regions where the surrogate has high uncertainty). In general, the next point is chosen by finding the global optimum $\mathbf{x}^{(m+1)} = \underset{\mathbf{x}}{\operatorname{argmax}}\, A(M, \mathbf{x})$.

Algorithm 1 Surrogate-based optimization

Require: budget B, surrogate model M, acquisition function A
1: Initialize $\mathbf{x}^{(1)}$ randomly and an empty set H
2: **for** $m = 1 : B$ **do**
3: $\quad y^{(m)} \leftarrow f(\mathbf{x}^{(m)}) + \epsilon$
4: $\quad H \leftarrow H \cup \{(\mathbf{x}^{(m)}, y^{(m)}))\}$
5: $\quad M \leftarrow$ fit surrogate model using H
6: $\quad \mathbf{x}^{(m+1)} \leftarrow \underset{\mathbf{x}}{\operatorname{argmax}}\, A(M, \mathbf{x})$
7: **end for**
8: **return** optimal $(\mathbf{x}^*, y^*) \in H$

Related Work

Although discrete problem structures are difficult to handle in black-box optimization, multiple approaches have been proposed. A survey by M. Zaefferer [24] presents different strategies for dealing with discrete structures in surrogate-based algorithms. The first strategy is the naive way by simply ignoring the discrete structure. Another strategy is to use inherently discrete models such as tree-based models [2, 12]. These models can however fail if the problem structure is too complex or if there are both discrete and continuous variables involved [24]. Lastly, discrete structures can be dealt with by using a certain mapping. Although this strategy does not apply directly to a surrogate model, a suitable mapping can make the problem easier. For example, encoding integer solutions with a binary representation can be easier for some regression models to handle.

There are also other strategies such as using problem-specific feature extraction or customizing the model. However, these violate the black-box assumption which is why we will not discuss them.

We now discuss several surrogate-based optimization algorithms that can solve the expensive discrete optimization problem in eq. (1) and that also have their code available online.

Bayesian optimization has a long history of success in expensive optimization problems [13], and has been applied in many domains such as chemical design and hyper-

parameter optimization for deep learning [9, 14]. It typically uses a Gaussian process as a surrogate to approximate the expensive objective. Several acquisition functions exist to guide the search, such as Expected Improvement, Upper Confidence Bound, or Thompson sampling [21], information-theoretic approaches such as Predictive Entropy Search [11], or simply the surrogate itself [5, 18]. Though Gaussian processes are typically used on continuous problems, they can be adapted for problems with discrete variables as well. The authors of [8] suggest three possible approaches, namely rounding to the nearest integer 1) when choosing where to evaluate the objective function, 2) when evaluating the objective function, or 3) inside the covariance function of the Gaussian process. The latter provides the best results but gives an acquisition function that is hard to optimize. The first option leads to the algorithm getting stuck by repeatedly evaluating the same points, although this can be circumvented by carefully balancing exploration and exploitation [17]. In this work, we will consider only the simpler second option, for which we do not need to modify any existing implementations.[1]

BOCS[2] [1] transforms the combinatorial problem into one that can be solved with semi-definite programming. It uses Thompson sampling as the acquisition function. However, it suffers from a large time complexity, which was only recently overcome by using a submodular relaxation called the PSR method[3] [6].

COMBO[4] [23] uses an efficient approximation of a Gaussian process with random features, together with Thompson sampling as the acquisition function. Though this gives increased efficiency, COMBO deals with discrete search spaces by iterating over all possible candidate solutions, which is only possible for small-dimensional problems.

HyperOpt[5] [2] makes use of a tree of Parzen estimators as the surrogate model. It can naturally deal with categorical or integer variables, and even with conditional variables that only exist if other variables take on certain values. The algorithm is known to perform especially well on hyperparameter tuning problems with hundreds of dimensions [3]. This is in sharp contrast with Bayesian optimization algorithms using Gaussian processes, which are commonly used on problems with less than 10 dimensions. A possible drawback for HyperOpt is that each dimension is modeled separately, i.e., no interaction between different variables is modeled. HyperOpt uses the Expected Improvement acquisition function.

SMAC[6] [12] is another surrogate-based algorithm that can naturally deal with integer variables. The main reason for this is that the surrogate model used in this algorithm is a random forest, which is an inherently discrete model. A point of critique for SMAC is that the random forests have worse predictive capabilities than Gaussian processes. Nevertheless, like HyperOpt, SMAC has been applied to problems with hundreds of dimensions [16]. SMAC uses the Expected Improvement acquisition function.

[1] We consider the implementation from https://github.com/fmfn/BayesianOptimization in this work, which uses the Upper Confidence Bound acquisition function.
[2] https://github.com/baptistar/BOCS
[3] https://github.com/aryandeshwal/Submodular_Relaxation_BOCS
[4] https://github.com/tsudalab/combo
[5] https://github.com/hyperopt/hyperopt
[6] https://github.com/automl/SMAC3

IDONE[7] [4] uses a linear combination of rectified linear units as its surrogate model. This is a continuous function, yet it has the special property that any local minimum of the model is located in a point where all variables take on integer values. This makes the method suitable for expensive discrete optimization problems, with the advantage that the acquisition function can be optimized efficiently with continuous solvers. IDONE uses the surrogate model itself as the acquisition function, but adds small perturbations to the optimum of the acquisition function to improve its exploration capabilities. Though the method is not as mature as SMAC or HyperOpt, it also has been applied to problems with more than 100 variables [4].

Benchmark problems

We present the four different benchmark problems that are used to compare the surrogate-based algorithms. The purpose of the benchmarks is to compare the discrete surrogate-based algorithms presented in the previous section and investigate which algorithms are most suited for which type of problem.

The benchmarks have been selected to include binary, categorical and ordinal decision variables but also different discrete structures such as sequential or graph-based structures. Since we assume that the evaluation of the objective functions is expensive, we perform the benchmark with a relatively strict budget of at most 500 evaluations. The objective function is evaluated once per iteration in Algorithm 1. Furthermore, we are testing on relatively large problem sizes, ranging from 44 up to 150 decision variables with search spaces of around $\sim 10^{50}$ possibilities. This range is interesting considering that Bayesian optimization using Gaussian processes is typically applied on problems with less than 10 variables.

On top of that, it has been shown that a large dimensionality reduces the importance of choosing a complicated acquisition function [18], which helps us doing a fair comparison between surrogates.

Moreover, we do an analysis of the performance of each algorithm where we limit the allowed time budget instead of the number of evaluations and simulate different evaluation times of the objective functions. The time budget includes both the total time to evaluate the objective function and the computation time of the optimization algorithm. Thus, it puts emphasis on the computation time of the algorithm in addition to their respective sample efficiency.

We present the four benchmark problems in detail below. Note that we present these problems in detail but that they are treated as black boxes by the optimization algorithms.

The Discrete Rosenbrock problem is a d-dimensional, non-convex function, with a curved valley that contains the global optimum defined by the following function:

$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(\mathbf{x}_{i+1} - \mathbf{x}_i^2)^2 + (1 - \mathbf{x}_i)^2] \tag{2}$$

[7] https://bitbucket.org/lbliek2/idone

where $\mathbf{x} \in \mathbb{Z}^d$. In the Rosenbrock problem, finding the valley is simple, but finding the global optimum $[1, 1, \ldots, 1]$ is not. As we are exploring discrete optimization problems, we consider a discrete variant of the problem such that only integer solutions are considered. We have $d = 49$ decision variables and each decision variable x_i is bounded by the range $[-5, 10]$. Thus, the problem's search space is in the order of 10^{59} candidate solutions. Lastly, the additive noise ϵ is normally distributed according to $N(\mu = 0, \sigma = 10^{-6})$.

The Weighted Max-Cut problem is an NP-hard graph cutting problem, defined as follows: For an undirected weighted graph $G = (V, E)$, a *cut* in G creates the subset $S \subseteq V$ and its complement $\overline{S} = V \backslash S$. Then $E(S, \overline{S})$, is defined as the set of edges that have one vertex in S and the other in $\overline{S}$. The Max-Cut problem is to find the cut that maximizes the weight of the edges in $E(S, \overline{S})$. The problem is encoded with a binary string $x \in \{0, 1\}^d$ where either $x_i = 0$ or $x_i = 1$ indicates if node i lies in S or $\overline{S}$ respectively.

For the following experiments, the MaxCut problem instances are randomly generated as weighted graphs, with d nodes, edge probability $p = 0.5$ and a uniformly distributed edge weight in the range $[0, 10]$. The graph generator is initialized with the same random seed for every run, ensuring that all experiments of a given problem size are performed on the same graph. On top of that, the additive noise ϵ added to each evaluation is following a standard normal distribution $N(\mu = 0, \sigma = 1)$. Lastly, we are using a graph with $d = 150$ nodes which means that the size of the problem's search space is $2^{150} \approx 10^{57}$.

The Perturbed Traveling Salesman is a variant of the well-known sequential graph problem where, given a number of cities and the distances between these cities, a shortest path needs to be found that visits all cities and returns to the starting city. We consider the asymmetric case with k cities where the distance between cities is not the same in both directions. Moreover, noise $\epsilon \sim U(0, 1)$ is added to each distance during evaluation. While the perturbation can cause issues for problem-specific solvers, it creates a good benchmark for black-box optimization algorithms. To ensure a robust solution, each proposed route is also evaluated 100 times and the worst-case objective value is returned. Furthermore, we will consider problem instance *ftv44*. This is an instance with 44 cities taken from TSPLIB [20], a library of problem instances for the traveling salesman problem. An instance with 44 cities is chosen to closely match the number of decision variables in the ESP problem which has a fixed number of 49 decision variables.

The problem is encoded as in [4]: after choosing a fixed origin city, there are $d = k - 2$ ordered decision variables x_i for $i = 1, \ldots, d$ such that $x_1 \in \{1, 2, \ldots, k - 1\}$ where each integer represents a city other than the origin city. Then, the next decision variable $x_2 \in \{1, ..., k - 2\}$ selects between the cities that were not yet visited. This is repeated until all cities have been chosen in some order. Since the last decision variable $x_d \in \{1, 2\}$ selects between the two remaining cities, we can deduce afterward the two remaining edges which closes the route since there is one last city to visit before returning to the origin city. Thus, the total number of possible sequences is given by $(d - 1)! \approx 6 \cdot 10^{52}$ for this instance.

The Electrostatic Precipitator problem is a real-world industrial optimization problem first published by Rehbach et al. [19]. The Electrostatic Precipitator (ESP) is a crucial component for gas cleaning systems. It is a large device that is used when solid particles need to be filtered from exhaust gases, such as reducing pollution in fossil fueled power plants. Before gas enters the ESP, it passes through a gas distribution system that controls the gas flow into the ESP. The gas flow is guided by configurable metal plates which blocks the airflow to a varying degree. The configuration of these plates inside the gas distribution system is vital for the efficiency of the ESP. However, it is non-trivial to configure this system optimally.

The objective function is computed with a computationally intensive fluid dynamics simulation, taking about half a minute of computation time every time a configuration is tested. There are 49 slots where different types of plates can be placed or be left empty. In total, there are 8 different options available per slot. This is formalized such that each integer-valued solution $\mathbf{x}$ is subject to the inequality constraint $0 \leq \mathbf{x}_i \leq 7$ for $i = 1, \ldots, 49$. This gives a large solution space in the order of 10^{44} possibilities.

Lastly, the problem has some ordinal structure where the decision variables decides between sizes of holes which are covering the plates. However, as an indication of the complex problem structure we have noted that changing any single variable does not affect the objective function.

Experiments

The goal of this section is to show a benchmark comparison between discrete and continuous surrogate-based algorithms on the discrete optimization problems of the previous section. The compared algorithms are HyperOpt and SMAC as two popular surrogate-based algorithms that make use of a discrete surrogate model if the search space is discrete, and Bayesian optimization as a popular surrogate-based algorithm for continuous problems. Though there exist several other algorithms that can deal with the discrete setting, these three are often used in practice because they are well established, can be used for a wide variety of problems, and have code available online. We also include IDONE in the comparisons as a surrogate-based algorithm that uses a continuous surrogate model but is designed for discrete problems, and random search is included as a baseline.

All experiments were run on the same Unix-based laptop with a Dual-Core Intel Core i5 2.7 GHz CPU and 8 GB RAM. Each algorithm attempted to solve the benchmarks 5 times. The allowed evaluation budget was 500 evaluations for all problems except the ESP problem where 100 evaluations were allowed instead due to it being more computationally expensive.

We are using the default hyperparameters for all algorithms, which are decided by their respective code libraries, with two exceptions. We change the SMAC algorithm to deterministic mode, since it otherwise evaluates the same point several times, which deteriorates its performance significantly. Besides that, the first five iterations of IDONE are random evaluations, which is similar to what happens in the other algorithms. The other algorithms start with their default number of random evaluations (which is 5 for Bayesian optimization and 3 for SMAC and HyperOpt), however for a fair comparison

we make sure that all of these initial random evaluations come from a uniform distribution over the search space. Unfortunately, more extensive hyperparameter tuning than stated above is too time-consuming for expensive optimization problems such as ESP.

In the following section we present the results from the benchmark comparison of the four surrogate-based optimization algorithms. The benchmark consists of the four problems which have varying discrete structures.

Results

In this section we describe the main results from comparing the algorithms on the discrete Rosenbrock, weighted Max-Cut, the travelling salesman and the ESP problems. Figure 2 shows the best average objective value found until a given iteration on each problem as well as their respective computation time. The computation time is the cumulative time up until iteration i which is required to perform the steps on line 5 and 6 in Algorithm 1. Furthermore, we also investigate how the algorithms perform if we introduce a time budget during optimization instead of constraining the number of evaluations.

Ordinal structures We start by comparing the results from the 49-dimensional discrete Rosenbrock problem. In Figure 2a, we see that Bayesian optimization (BO) is the only algorithm that comes close to the optimal objective value of zero. The other algorithms are not performing as well, where HyperOpt (HO) gets the closest to BO. Given that the problem is in fact a discrete version of an inherently continuous problem with ordinal variables, this can be considered to be well suited for continuous model regression. On the other hand, IDONE also uses a continuous surrogate, but it does not perform as well as BO. A possible explanation is that IDONE is less flexible since it is a piece-wise linear model.

To investigate the quality of the surrogates from both BO and IDONE, we visualize their surfaces in Figure 1 for the 2-dimensional case of Rosenbrock. The Gaussian process from BO (which uses a Matérn 5/2 kernel in this case) predicts a smoother surface than IDONE which appears more rugged and uneven. Overall, BO looks more similar to the objective ground truth. We can argue that this is why BO performs well while IDONE does not. BO is likely suitable for the discrete Rosenbrock problem since the problem has an underlying continuous structure with ordinal variables. Meanwhile, this structure could be too complex for the piece-wise linear surrogate in IDONE.

However, we are interested in investigating problems which do not necessarily have a clear continuous structure. Thus, we look at the ESP problem which also happens to have some ordinal structure. The results from this problem are found in Figure 2c. It shows a more even performance among the algorithms compared to the Rosenbrock problem, although BO still returns the best objective on average. This is closely followed by both SMAC and HO, while IDONE is doing worse than random search.

Based on the results from these two problems, it appears that BO works well on ordinal structures. However, this does not seem to hold true for all continuous surrogates considering the performance of IDONE. Still, the naive approach with BO outperforms the other state-of-the-art discrete algorithms on the problems that we have discussed so

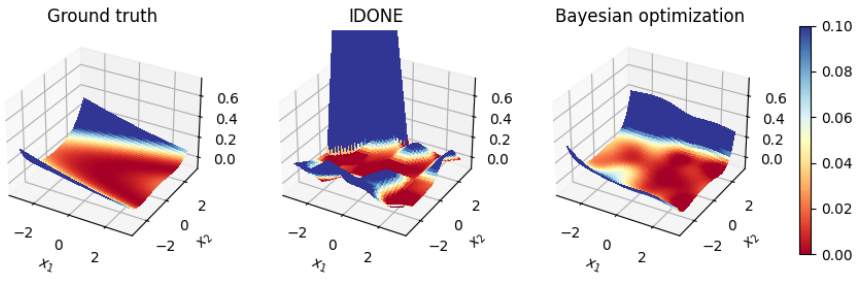
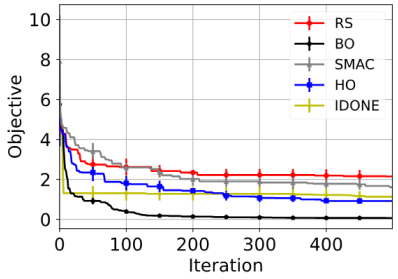


Fig. 1: Visualization of continuous surrogates that approximate the two-dimensional Rosenbrock, namely the linear combination of ReLUs from IDONE and Gaussian processes from BO. These models were picked based on the best performance from 15 different runs with 50 evaluations each. HyperOpt and SMAC are not visualized since this is not supported by their respective code libraries.

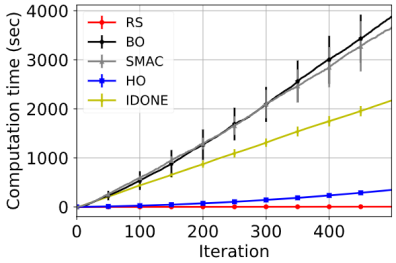
far. This is actually in line with experimental results from [8] on small problems (up to 6 dimensions) with both discrete and continuous parameters, though it was not the main conclusion of the authors. The difference with our work is that we consider purely discrete problems of higher dimensions, from a real-life application, and we include IDONE in the comparison.

Binary structures We will now consider a graph problem, that is the weighted Max-Cut problem. From the results in Figure 2e, we notice that BO clearly outperforms all other algorithms. Meanwhile, IDONE is the second best, followed by SMAC and then HO which performs worse than random search. Compared to the other problems that we have seen so far, a major difference is the binary decision variables in the Max-Cut problem. We use this to frame our hypothesis, namely that the good performance of BO on the Max-Cut problem is due to the binary structure of the problem.

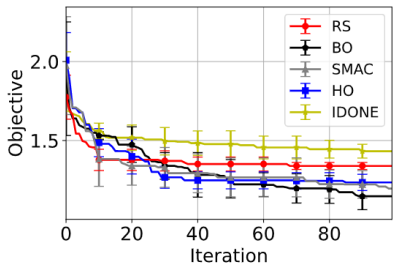
To investigate this hypothesis, we perform an additional experiment by encoding the 49-dimensional, discrete Rosenbrock with binary variables and compare this with the previous results from Figure 2a. The ordinary problem has 49 integer decision variables which lie in the range $[-5, 10]$, this is converted into a total of 196 binary decision variables for the binary-encoded version. Table 1 shows the performance of the algorithms on the binary-encoded, discrete Rosenbrock. Although BO is performing worse on the binarized Rosenbrock, it is still performing the best compared to the other algorithms, even though both SMAC and IDONE perform better on the binarized problem. Thus, we could argue that the binary representation of the Max-Cut problem can not explain



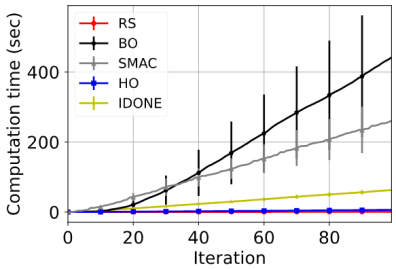
(a) Best average objective value versus iteration on the 49-dimensional discrete Rosenbrock.



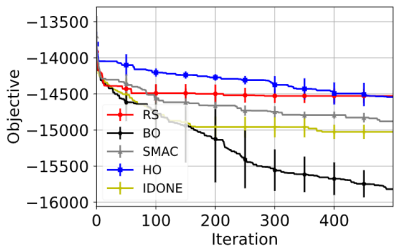
(b) Average computation time versus iteration on the 49-dimensional discrete Rosenbrock.



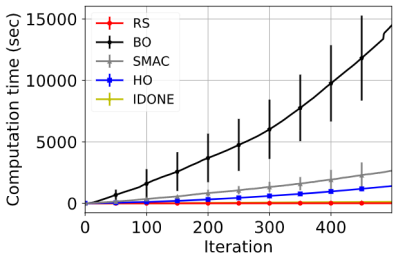
(c) Best average objective value versus iteration on the ESP problem.



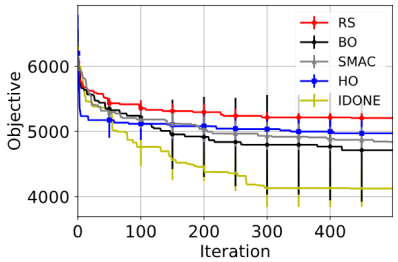
(d) Average computation time versus iteration on the ESP problem.



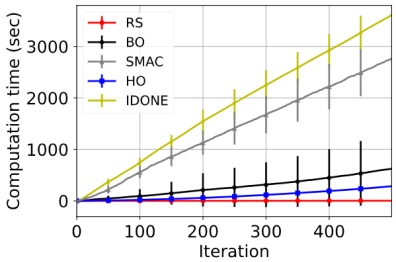
(e) Best average objective value versus iteration on the 150-dimensional weighted Max-Cut.



(f) Average computation time versus iteration on the 150-dimensional weighted Max-Cut.



(g) Best average objective value versus iteration on the TSP with 44 cities.



(h) Average computation time versus iteration on the TSP with 44 cities.

Fig. 2: Comparison of objective value and computation time of Bayesian optimization (BO), SMAC, IDONE, HyperOpt (HO) and random search (RS) on four different benchmark problem. An average is computed from 5 runs and the standard deviation is plotted as the error. The objective value has been negated for Max-Cut since the maximization problem has been turned into a minimization problem. The evaluation budget was 500 evaluations for all problems except the ESP problem which was limited to 100 evaluations due to it being more computationally expensive.

why BO performs well on this problem. There is a possible argument that the binary variables might cause less rounding-off errors since the range of values is simply zero to one with a threshold in the middle. However, a counter-argument is that such a large number of decision variables is typically not well-suited for Gaussian processes regression. This is also indicated by the large computation time of BO on the Max-Cut, see Figure 2f, compared to the other problems as well.

Sequential structures Even though TSP is a graph problem like the Max-Cut problem, there is an important difference. TSP has a sequential structure since the decision variables select an ordering that directly affects the objective value. Moreover, the encoding of the problem, as described in the "Benchmarks problems" section, causes strong interactions between adjacent decision variables.

We continue by looking at the results from TSP in Figure 2g. BO is now outperformed by IDONE even though it still performs better than SMAC and HO on average, although BO has a large variance on this problem. We suspect that the sequential structure is well-suited for IDONE, as it explicitly fits some of its basis functions with adjacent variables in the input vector $(x_1, x_2, \ldots, x_d)$ [4].

To investigate whether this is the case, we test what happens when the order of the decision variables are re-shuffled in TSP such that the sequential structure is removed. This is done by adding to the objective function a mapping that changes the order of the variables in the input vector $(x_1, x_2, \ldots, x_d)$ to a fixed arbitrarily chosen order. From Table 2 we see that IDONE performs worse without the original sequential structure. At the same time, the other algorithms show no large significant difference. However, IDONE returns the best objective on average both with and without shuffling the order of variables. The large variance on BO makes it more difficult to draw any strong conclusions, but since IDONE also uses a continuous surrogate model, we can still conclude that continuous surrogates perform better than the discrete counterparts on this problem.

Algorithm	Non-binary	Binary
BO	**0.067 (0.021)**	**0.37 (0.038)**
SMAC	1.61 (0.18)	1.28 (0.29)
HyperOpt	0.91 (0.13)	0.94 (0.14)
IDONE	1.13 (0.20)	0.61 (0.038)

Table 1: Comparison of results on the 49-dimensional discrete Rosenbrock with and without binary encoding of the decision variables. The final average objective value from 5 runs is presented after 500 evaluations with the standard deviation in parenthesis. The lowest objective value is marked as bold in each column.

Algorithm	Non-shuffled	Shuffled
BO	4713.2 (789.2)	4898.0 (292.4)
SMAC	4841.8 (184.7)	4784.9 (302.7)
HyperOpt	4971.9 (256.5)	4871.8 (221.9)
IDONE	**4122.8 (279.8)**	**4556.4 (175.7)**

Table 2: Comparison of TSP with 44 cities when the input has a sequential structure versus that decision variables' position have been shuffled. The final average objective value from 5 runs is presented after 500 evaluations with the standard deviation in parenthesis. The lowest objective value is marked as bold in each column.

Taking computation time into consideration Although BO performs well on the benchmark comparisons, we are noticing that it is more expensive with respect to compute time compared to the other surrogate-based methods. Figures 2b, 2d, 2f and 2h show the cumulative time on the problems.

In general, BO requires a vast amount of time compared to the other algorithms, especially on Max-Cut where the computations took one to two minutes per iteration. This is not surprising considering that regression with Gaussian processes is computationally intensive: its complexity grows as $O(n^3)$ where n are the number of observations [21]. This can be a big drawback if the evaluation time of the objective function is relatively small.

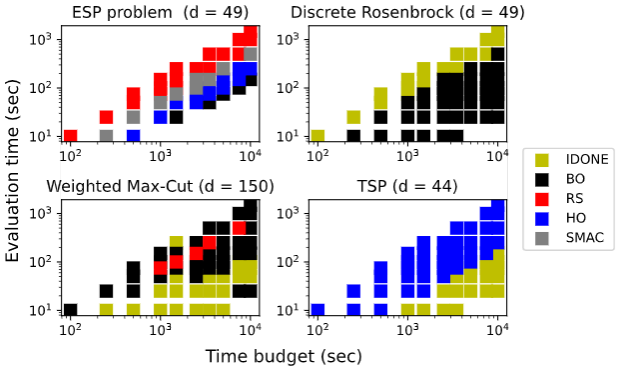


Fig. 3: The best algorithm on average when given a time budget and a fixed evaluation time of the objective function. The results are simulated by adding an artificial evaluation time after running the experiments. For the ESP problem, the actual evaluation time was about $3 \cdot 10^1$ seconds for each function evaluation. The time budget includes both the evaluation time and the compute time of the algorithms.

Meanwhile, the other algorithms share similar computation times which are often less than one second. The only exception is for IDONE which requires more computation time on TSP, see Figure 2h.

So far, we have only considered experiments that restrict the number of evaluations. But in real-life applications, the computation time of an algorithm can be important to take into consideration when limited with some given time budget as well. In particular, the large computation time of BO motivates the question whether it would still perform well under a constrained time budget instead. By keeping track of both the evaluation times of the objective functions, as well as the computation time spent by the algorithms at every iteration, we can investigate the performance of the algorithms in different situations. We artificially adjust the evaluation time in the experiments from Figure 2 to simulate the cost of the objective function. The evaluation time ranges from 10^1 to $1.5 \cdot 10^3$ seconds. Similarly, the time budget varies between 10^2 and 10^4 seconds.

Figure 3 displays which algorithm performs best on average for each problem, depending on the evaluation time and time budget. Notice that the lower triangular shape is caused by the fact that the time budget must be larger than the evaluation time. To ensure a fair comparison, we only present the algorithm with the best final average objective value if the maximum number of evaluations from the previous benchmark experiments was not exceeded within the allocated time budget for all algorithms.

For the ESP problem, the results are mixed. It seems like the best algorithm varies between BO, HO, SMAC and even random search, depending mostly on the ratio between the time budget and the evaluation time. For example, random search performs best when the evaluation time is around the order of 10^1 smaller than the time budget which gives relatively few evaluations. Meanwhile, BO performs best with a much larger ratio. On the discrete Rosenbrock benchmark, BO is clearly the best in almost all cases. The only exception is when the ratio between evaluation time and time budget is very small, in which case IDONE performs better. For the weighted Max-Cut, on the other hand, we notice the opposite of what we see with the Rosenbrock benchmark. Thus, it seems like the growth in compute time of BO, see Figure 2f, sometimes outweighs its good performance which we noted earlier when only taking an evaluation budget into consideration. Lastly, we see that IDONE and HyperOpt outperform other algorithms on TSP when constrained by a time budget.

This experiment gives a better picture of the performance of each algorithm, especially if we may consider it to be more realistic by taking time constraints into consideration. Thus, the experiment from Figure 3 is a good complement to our benchmark comparison. In the following and last section, we summarize the conclusions that can be drawn from all of the above experiments.

Conclusion and Future Work

Based on the results from the benchmark comparison, we can show that the use of continuous surrogate models is a valid approach for expensive, discrete black-box optimization. Moreover, we give insight into what discrete problem structures are well-suited for the different methods.

We have shown that Bayesian optimization (BO) performs better than discrete state-of-the-art algorithms on the four tested, high-dimensional benchmarks problem with either ordinal, sequential or binary structures. IDONE, another continuous surrogate-based algorithm designed for discrete problems, outperforms BO on the benchmark with a sequential structure, but not on the three other benchmarks.

In addition, we have investigated how the different algorithms deal with the different problem structures. Firstly, ordinal structures appear suitable for BO, especially if the objective function has an underlying continuous structure such as the discrete Rosenbrock benchmark. For binary structures, we noticed that BO is negatively affected by binary variables, while IDONE and SMAC benefited from this transformation. However, BO still returned the best solution on the binary Max-Cut problem, even though a big drawback was its computation time. Lastly, we have seen that IDONE outperforms the other algorithms on a problem with sequential decision variables, even after negatively affecting it by changing the ordering.

We also investigated the different algorithms under different time constraints by artificially changing the function evaluation times of the different benchmark problems. For lower time budgets, BO is held back by its large compute time in some cases. Even though BO is a time-intensive method, it mostly showed competitive performance when the evaluation time was relatively low and the time budget high, except for the binary Max-Cut problem. IDONE, HyperOpt, SMAC, and even random search all had specific problems and time budgets where they outperformed other algorithms. Lastly, based on our results, discrete surrogate-based methods could be more relevant in the setting with a limited time budget, in contrast to only limiting the number of evaluations.

Finally, we state some open questions which remain to be answered about continuous surrogates in the topic of expensive, discrete black-box optimization. Considering that we looked at a naive approach of BO, it is still an open question how the more advanced discrete BO variations would fare in the framework where time budgets and function evaluations times are taken into account like in this paper. This same framework would also lead to interesting comparisons between surrogate-based algorithms and other black-box algorithms such as local search or evolutionary algorithms, which are better suited for cheap function evaluations. It also remains unclear why BO performs best on the binary Max-Cut benchmark even though it is negatively affected by binary structures on the Rosenbrock function. Finally, it would be of great practical value if one could decide on the best surrogate-based algorithm in advance, given the time budget and evaluation time of a real-life optimization problem. This research is a first step in that direction.

Acknowledgments

This work is part of the research programme Real-time data-driven maintenance logistics with project number 628.009.012, which is financed by the Dutch Research Council (NWO). The authors would also like to thank Arthur Guijt for helping with the python code.

References

1. Baptista, R., Poloczek, M.: Bayesian optimization of combinatorial structures. In: International Conference on Machine Learning. pp. 471–480 (2018)
2. Bergstra, J., Yamins, D., Cox, D.D.: Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms. In: Proceedings of the 12th Python in science conference. pp. 13–20 (2013)
3. Bergstra, J., Yamins, D., Cox, D.D.: Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th International Conference on Machine Learning. Jmlr (2013)
4. Bliek, L., Verwer, S., de Weerdt, M.: Black-box combinatorial optimization using models with integer-valued minima. arXiv preprint arXiv:1911.08817 (2019)
5. De Ath, G., Everson, R.M., Rahat, A.A., Fieldsend, J.E.: Greed is good: Exploration and exploitation trade-offs in Bayesian optimisation. arXiv preprint arXiv:1911.12809 (2019)
6. Deshwal, A., Belakaria, S., Doppa, J.R.: Scalable combinatorial Bayesian optimization with tractable statistical models. arXiv preprint arXiv:2008.08177 (2020)

7. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. arXiv preprint arXiv:1808.05377 (2018)
8. Garrido-Merchán, E.C., Hernández-Lobato, D.: Dealing with categorical and integer-valued variables in Bayesian optimization with Gaussian processes. Neurocomputing **380**, 20–35 (2020)
9. Griffiths, R.R., Hernández-Lobato, J.M.: Constrained Bayesian optimization for automatic chemical design. arXiv preprint arXiv:1709.05501 (2017)
10. Hernández-Lobato, J.M., Gonzalez, J., Martinez-Cantin, R.: NIPS workshop on Bayesian optimization. https://bayesopt.github.io/, accessed 22-08-2020
11. Hernández-Lobato, J.M., Hoffman, M.W., Ghahramani, Z.: Predictive entropy search for efficient global optimization of black-box functions. In: Advances in neural information processing systems. pp. 918–926 (2014)
12. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: International conference on learning and intelligent optimization. pp. 507–523. Springer (2011)
13. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. Journal of Global optimization **13**(4), 455–492 (1998)
14. Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F.: Fast Bayesian optimization of machine learning hyperparameters on large datasets. In: Artificial Intelligence and Statistics. pp. 528–536 (2017)
15. Korovina, K., Xu, S., Kandasamy, K., Neiswanger, W., Poczos, B., Schneider, J., Xing, E.: Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In: International Conference on Artificial Intelligence and Statistics. pp. 3393–3403. PMLR (2020)
16. Lindauer, M., Hutter, F.: Warmstarting of model-based algorithm configuration. arXiv preprint arXiv:1709.04636 (2017)
17. Luong, P., Gupta, S., Nguyen, D., Rana, S., Venkatesh, S.: Bayesian optimization with discrete variables. In: Australasian Joint Conference on Artificial Intelligence. pp. 473–484. Springer (2019)
18. Rehbach, F., Zaefferer, M., Naujoks, B., Bartz-Beielstein, T.: Expected improvement versus predicted value in surrogate-based optimization. arXiv preprint arXiv:2001.02957 (2020)
19. Rehbach, F., Zaefferer, M., Stork, J., Bartz-Beielstein, T.: Comparison of parallel surrogate-assisted optimization approaches. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 1348–1355. GECCO '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3205455.3205587
20. Reinelt, G.: TSPlib. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html, accessed 31-07-2020
21. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N.: Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE **104**(1), 148–175 (2016)
22. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems. pp. 2951–2959 (2012)
23. Ueno, T., Rhone, T.D., Hou, Z., Mizoguchi, T., Tsuda, K.: Combo: An efficient Bayesian optimization library for materials science. Materials discovery **4**, 18–21 (2016)
24. Zaefferer, M.: Surrogate Models For Discrete Optimization Problems. Ph.D. thesis, Technischen Universität Dortmund (2018)

Comparing Correction Methods to Reduce Misclassification Bias

Kevin Kloos[1,5][0000−0001−6980−4259], Quinten Meertens[3,4,5][0000−0002−3485−8895], Sander Scholtus[5][0000−0002−8316−8938], and Julian Karch[2][0000−0002−1625−2822]

[1] Mathematical Institute, Leiden University, the Netherlands
[2] Institute of Psychology, Leiden University, the Netherlands
[3] Leiden Centre of Data Science, Leiden University, the Netherlands
[4] Center for Nonlinear Dynamics in Economics and Finance, University of Amsterdam, the Netherlands
[5] Statistics Netherlands, The Hague, the Netherlands †

Abstract. When applying supervised machine learning algorithms to classification, the classical goal is to reconstruct the true labels as accurately as possible. However, if the predictions of an accurate algorithm are aggregated, for example by counting the predictions of a single class label, the result is often still statistically biased. Implementing machine learning algorithms in the context of official statistics is therefore impeded. The statistical bias that occurs when aggregating the predictions of a machine learning algorithm is referred to as misclassification bias. In this paper, we focus on reducing the misclassification bias of binary classification algorithms by employing five existing estimation techniques, or estimators. As reducing bias might increase variance, the estimators are evaluated by their mean squared error (MSE). For three of the estimators, we are the first to derive an expression for the MSE in finite samples, complementing the existing asymptotic results in the literature. The expressions are then used to compute decision boundaries numerically, indicating under which conditions each of the estimators is optimal, i.e., has the lowest MSE. Our main conclusion is that the calibration estimator performs best in most applications. Moreover, the calibration estimator is unbiased and it significantly reduces the MSE compared to that of the uncorrected aggregated predictions, supporting the use of machine learning in the context of official statistics.‡

Keywords: Bias Correction · Misclassification Bias · Supervised Machine Learning · Classification · Official Statistics

†Corresponding authors: k.kloos@cbs.nl, q.a.meertens@uva.nl

‡The views expressed in this paper are those of the authors and do not necessarily reflect the policy of Statistics Netherlands. The authors would like to thank Arnout van Delden and three anonymous referees for their useful comments on previous versions of this paper.

1 Introduction

Currently, many researchers in the field of official statistics are examining the potential of machine learning algorithms. A typical example is estimating the proportion of houses in the Netherlands having solar panels, by employing a machine learning algorithm trained to classify satellite images [3]. However, as long as the algorithm's predictions are not error-free, the estimate of the relative occurrence of a class, also known as the *base rate*, can be biased [17,18]. This fact is also intuitively clear: if the number of false positives does not equal the number of false negatives, then the estimate of the base rate is biased, even if the false positive rate and false negative rate are both small. The statistical bias that occurs when aggregating the predictions of a machine learning algorithm is referred to as *misclassification bias* [5].

Misclassification bias occurs in a broad range of applications, including official statistics [13], land cover mapping [12], political science [9,21], and epidemiology [8]. The objective in each of these applications is to minimize a loss function at the level of aggregated predictions, in contrast to minimizing a loss function at the level of individual predictions. Within the field of machine learning, learning with that objective is referred to as quantification learning; see [6] for a recent overview. In quantification learning, the idea is not to train a classifier at all, but to directly estimate the base rate from the feature distribution. A drawback of that approach is that relatively large training and test datasets are needed to optimize hyperparameters and to obtain accurate estimates of the accuracy of the prediction, respectively. In the applications referred to before, labelled data are often expensive to obtain and therefore scarce. Hence, in this paper, we focus on what is referred to as quantifiers based on corrected classifiers [6]. In short, it entails that we first aggregate predictions of classification algorithms and then correct the aggregates in order to reduce misclassification bias.

In the literature on measurement error, several methods have been proposed to reduce misclassification bias when aggregating categorical data that is prone to measurement error; see [11] for a technical discussion and [1] for a more recent overview. Based on that literature, we propose a total of five estimators for the base rate that can be derived from the confusion matrix of a classification algorithm. As reducing bias might increase variance, the estimators are evaluated by their mean squared error (MSE). To the best of our knowledge, for three of the five estimators, only asymptotic expressions for the MSE are ever presented in the literature. In this paper, we derive the expressions for the MSE for finite datasets. As a first step, we restrict ourselves to binary classification problems. Nonetheless, we believe that the same proof strategies may be used for multi-class classification problems. The expressions for the MSE enable a theoretical comparison of the five estimators for finite datasets. It allows us, for the first time, to make solid recommendations on how to employ classification algorithms in official statistics and other disciplines interested in aggregate statistics.

The remainder of the paper is organized as follows. First, in Section 2, the five estimators are formally introduced and the mathematical expressions for their MSEs are presented. The derivations are included in the appendix. Then, in

Section 3, the decision boundaries are numerically derived. We can indicate under which condition, like the sensitivity and specificity of the learning algorithm and the size of the test set, each of the estimators has the lowest MSE. Finally, in Section 4, we draw our main conclusion and discuss directions for future research.

2 Methods

Consider a *target population* of N objects and assume that the objects can be separated into two classes. One of the two classes is the *class of interest*. We refer to the relative occurrence of the class of interest in the target population as the *base rate* and we denote that parameter by α. In the example mentioned in Section 1, the objects are houses in the Netherlands and the two classes are whether or not the house has solar panels on the roof [3]. The class of interest is having solar panels and hence α indicates the relative frequency of houses in the country having solar panels.

We assume that the true classifications are only known for objects in a small simple random sample of the target population. In the applications that we consider, these classifications are obtained by manual inspection of the objects in that sample. Objects that belong to the class of interest receive class label 1, the other objects receive class label 0. Then, the sample is split randomly into a training set and a test set. As usual, the training set is used for model selection through cross-validation and is then used to train the selected model. We will consider the result of that part of the process as given. The test set is used to estimate the classification performance of the trained algorithm, which we will discuss in more detail below. Finally, the classification algorithm is applied on the entire target population (minus the small random sample, but we will neglect that small difference) resulting in a predicted label for each object.

As we will encounter in Subsection 2.2, simply computing the relative occurrence of objects predicted to belong to the class of interest will result in a biased estimate of α. That bias is referred to as *misclassification bias* [4]. In this section, five estimators for the base rate parameter α are formally introduced, many of which have been proposed decades ago; see [11] for an extensive discussion. We summarize the formulas for bias and variance that can be found in the literature and complement them with our own derivations.

In order to correct for misclassification bias, we need estimates of the algorithm's (mis)classification probabilities. Following [20], we assume that misclassifications are independent across objects and that the (mis)classification probabilities are the same for each object, conditional on their true class label. With this classification-error model in mind, we denote the probability that the algorithm predicts an object of class 0 correctly by p_{00} and we define p_{11} analogously. Observe that p_{11} and p_{00} correspond to the algorithm's sensitivity and specificity, respectively. The *confusion matrix* $\mathbf{P}$ is then defined as follows:

$$\mathbf{P} = \begin{pmatrix} p_{00} & 1 - p_{00} \\ 1 - p_{11} & p_{11} \end{pmatrix}. \tag{1}$$

Table 1: Contingency tables for test set (left) and target population (right)

(a)

		Estimated class		
		0	1	Tot
True class	0	n_{00}	n_{01}	n_{0+}
	1	n_{10}	n_{11}	n_{1+}
	Tot	n_{+0}	n_{+1}	n

(b)

		Estimated class		
		0	1	Tot
True class	0	N_{00}	N_{01}	N_{0+}
	1	N_{10}	N_{11}	N_{1+}
	Tot	N_{+0}	N_{+1}	N

The classification probabilities p_{00} and p_{11} are not known, but will be estimated using the test set. We write n for the size of the test set and introduce the notation n_{ij} and N_{ij} as depicted in Table 1. The classification probabilities are then estimated without bias by $\hat{p}_{00} = n_{00}/n_{0+}$ and $\hat{p}_{11} = n_{11}/n_{1+}$. (Here, the assumption is needed that the test set is a simple random sample from the target population.) Furthermore, the base rate α for the target population is defined formally as $\alpha = N_{1+}/N$.

Finally, we make the following technical assumptions. We assume that the algorithm is not perfect in predicting either of the classes, but that it is better than guessing for both of the classes, i.e., we assume that $0.5 < p_{ii} < 1$. Because the test set is a small (i.e., $n \ll N$) simple random sample from the population, n_{0+} may be assumed to follow a $Bin(n, \alpha)$-distribution, since α is considered fixed. Moreover, the classification-error model that we assume implies that the elements in the rows in Table 1, conditional on the corresponding row total, follow a binomial distribution as well, with the corresponding classification probability as success probability. For example, to name just two out of the eight entries, $n_{00} \mid n_{0+} \sim Bin(n_{0+}, p_{00})$ and $N_{10} \mid N_{1+} \sim Bin(N_{1+}, 1-p_{11})$. Last, the assumption $n \ll N$ justifies our ultimate technical assumption, which is that the estimators for the entries in $\mathbf{P}$ based on the test set on the one hand and estimators for α based only on the predicted class labels for the target population on the other hand, are independent random variables.

2.1 Baseline estimator - random sample

The baseline estimator for α is the proportion of data points in the test dataset for which the observed class label is equal to 1. The baseline estimator will be denoted by $\hat{\alpha}_a$. Under the assumptions discussed above, it is immediate that $\hat{\alpha}_a$ is an unbiased estimator for α, i.e.:

$$B\left[\hat{\alpha}_a\right] = 0. \tag{2}$$

Since we have assumed that the size n of the test dataset is much smaller than the size N of the population data, we may approximate the distribution of $n\hat{\alpha}_a$ by a binomial distribution with success probability α. The variance, and hence the MSE, of $\hat{\alpha}_a$ is then given by

$$MSE\left[\hat{\alpha}_a\right] = V\left[\hat{\alpha}_a\right] = \frac{\alpha(1-\alpha)}{n}. \tag{3}$$

This MSE will serve as the baseline value for the other estimators we discuss.

2.2 Classify and count

When applying a trained machine learning algorithm on new data, we may simply count the number of data points for which the predicted class equals 1. The resulting estimator for α, which we will denote by $\hat{\alpha}^*$, is referred to as the 'classify-and-count' estimator, see [6]. In general, the classify-and-count estimator is (strongly) biased, and has almost zero variance. More specifically,

$$E\left[\hat{\alpha}^*\right] = \alpha p_{11} + (1-\alpha)(1-p_{00}), \tag{4}$$

and hence

$$B\left[\hat{\alpha}^*\right] = \alpha(p_{11}-1) + (1-\alpha)(1-p_{00}), \tag{5}$$

which is zero only if the point (p_{00}, p_{11}) lies on the line through $(1-\alpha, \alpha)$ and $(1,1)$ in $\mathbb{R}^2$, as shown in [17]. The variance of the classify-and-count estimator is derived in [2] and equals

$$V\left[\hat{\alpha}^*\right] = \frac{\alpha p_{11}(1-p_{11}) + (1-\alpha)p_{00}(1-p_{00})}{N}. \tag{6}$$

If the population size N is large, the variance of $\hat{\alpha}^*$ is low. In some literature, this low variance is misinterpreted as high accuracy, by claiming intuitively that the large size of the dataset implies that the noise cancels out (cf. [16]). However, the nonzero bias is neglected in such arguments. Therefore, we are interested in the MSE because it considers both bias and variance. It equals

$$MSE\left[\hat{\alpha}^*\right] = \Big[\alpha(p_{11}-1) + (1-\alpha)(1-p_{00})\Big]^2 + O\left(\frac{1}{N}\right). \tag{7}$$

Here and below, the notation $O(1/x)$ indicates a remainder term that, for sufficiently large values of $x > 0$, is always contained inside an interval $(-C/x, C/x)$ for some constant $C > 0$; see, e.g., [19, p. 147]. Observe how, in general, the MSE does not converge to 0 as N tends to ∞.

2.3 Subtracting estimated bias

Knowing that the classify-and-count estimator $\hat{\alpha}^*$ is biased (see (5)), we may attempt to estimate that bias and subtract it from $\hat{\alpha}^*$. As briefly mentioned in [17], we may estimate that bias by the plug-in estimator, that is, we substitute the unknown quantities in Equation (5) by their estimates. More precisely, the bias is estimated as

$$\widehat{B}\left[\hat{\alpha}^*\right] = \hat{\alpha}^*(\hat{p}_{00} + \hat{p}_{11} - 2) + (1-\hat{p}_{00}), \tag{8}$$

in which the estimators $\hat{p}_{00}$ and $\hat{p}_{11}$ are based on the test dataset. The resulting estimator $\hat{\alpha}_b$ for α equals

$$\hat{\alpha}_b = \hat{\alpha}^* - \widehat{B}\left[\hat{\alpha}^*\right] = \hat{\alpha}^*(3 - \hat{p}_{00} - \hat{p}_{11}) - (1-\hat{p}_{00}). \tag{9}$$

To the best of our knowledge, the bias and variance of the estimator $\hat{\alpha}_b$ have not been published in the scientific literature. Therefore, we have derived both, up to terms of order $1/n^2$, yielding the following result.

Theorem 1. *The bias of $\hat{\alpha}_b$ as estimator for α is given by*

$$B\left[\hat{\alpha}_b\right] = (1 - p_{00})(2 - p_{00} - p_{11}) - \alpha(p_{00} + p_{11} - 2)^2. \tag{10}$$

The variance of $\hat{\alpha}_b$ equals

$$\begin{aligned} V\left[\hat{\alpha}_b\right] = {} & \frac{\left[\alpha(p_{00} + p_{11} - 1) - p_{00}\right]^2 p_{00}(1 - p_{00})}{n(1 - \alpha)} \left(1 + \frac{\alpha}{n(1 - \alpha)}\right) \\ & + \frac{\left[\alpha(p_{00} + p_{11} - 1) + (1 - p_{00})\right]^2 p_{11}(1 - p_{11})}{n\alpha} \left(1 + \frac{1 - \alpha}{n\alpha}\right) \\ & + O\left(\max\left[\frac{1}{n^3}, \frac{1}{N}\right]\right). \end{aligned} \tag{11}$$

Proof. See the Appendix.

In particular, Theorem 1 implies that $B\left[\hat{\alpha}_b\right] = (2 - p_{00} - p_{11})B\left[\hat{\alpha}^*\right]$, compare Equations (10) and (5). Hence, $|B\left[\hat{\alpha}_b\right]| \leq |B\left[\hat{\alpha}^*\right]|$, because $1 < p_{00} + p_{11} < 2$.

2.4 Misclassification probabilities

Let $\mathbf{P}$ be the row-normalized confusion matrix of the machine learning algorithm that we have trained, as defined in (1). That is, entry p_{ij} is the probability that the algorithm predicts class j for a data point that belongs to class i. The probabilities p_{ij} are referred to as misclassification probabilities. In the binary setting, we write $\boldsymbol{\alpha}$ for the column vector $(1 - \alpha, \alpha)^T$ (similarly for $\hat{\boldsymbol{\alpha}}^*$). Under the assumption that the probabilities p_{ij} are identical for each data point, we obtain the expression $E[\hat{\boldsymbol{\alpha}}^*] = \mathbf{P}^T\boldsymbol{\alpha}$. If the true values of all entries p_{ij} of $\mathbf{P}$ were known and if $p_{00} + p_{11} \neq 1$, then $\hat{\boldsymbol{\alpha}}_p = (\mathbf{P}^T)^{-1}\hat{\boldsymbol{\alpha}}^*$ would be an unbiased estimator for α. Using the plug-in estimator $\hat{\mathbf{P}}$ for $\mathbf{P}$, estimated on the test set, the following estimator for α is obtained:

$$\hat{\alpha}_p = \frac{\hat{\alpha}^* + \hat{p}_{00} - 1}{\hat{p}_{00} + \hat{p}_{11} - 1}. \tag{12}$$

It is known that the estimator $\hat{\alpha}_p$ is consistent (asymptotically unbiased) for α, see [1]. In [7], the variance of this estimator is analysed for an arbitrary number of classes. For the binary case, a simple analytic expression for the bias and variance of $\hat{\alpha}_p$ for finite datasets has not been given, as far as we know. Therefore, we have derived the bias and variance for finite datasets, yielding the following result.

Theorem 2. *The bias of $\hat{\alpha}_p$ as estimator for α is given by*

$$B\left[\hat{\alpha}_p\right] = \frac{p_{00} - p_{11}}{n(p_{00} + p_{11} - 1)} + O\left(\frac{1}{n^2}\right). \tag{13}$$

The variance of $\hat{\alpha}_p$ is given by

$$V[\hat{\alpha}_p] = \frac{(1-\alpha)p_{00}(1-p_{00})\left[1 + \frac{\alpha}{n(1-\alpha)}\right] + \alpha p_{11}(1-p_{11})\left[1 + \frac{1-\alpha}{n\alpha}\right]}{n(p_{00} + p_{11} - 1)^2} + O\left(\max\left[\frac{1}{n^2}, \frac{1}{N}\right]\right). \tag{14}$$

Proof. See the Appendix.

2.5 Calibration probabilities

Let $\mathbf{C}$ be the column-normalized confusion matrix of the machine learning algorithm that we have trained. That is, entry c_{ij} is the probability that the true class of a data point is j given that the algorithm has predicted class i. The probabilities c_{ij} are referred to as calibration probabilities [11]. The first element of the vector $\mathbf{C}\hat{\boldsymbol{\alpha}}^*$ is an unbiased estimator for $\boldsymbol{\alpha}$, if $\mathbf{C}$ is known.

Using the plug-in estimator $\hat{\mathbf{C}}$ for $\mathbf{C}$, which is estimated on the test dataset analogously to $\hat{\mathbf{P}}$, the following estimator $\hat{\alpha}_c$ for α is obtained:

$$\hat{\alpha}_c = \hat{\alpha}^* \frac{n_{11}}{n_{+1}} + (1 - \hat{\alpha}^*)\frac{n_{10}}{n_{+0}}, \tag{15}$$

in which each n_{ij} and n_{+j} should be considered as random variables. It has been shown that $\hat{\alpha}_c$ is a consistent estimator for α [1]. Under the assumptions we have made in this paper, it can be shown that $\hat{\alpha}_c$ is in fact an unbiased estimator for α. To the best of our knowledge, we are also the first to give an approximation (up to terms of order $1/n^2$) of the variance of $\hat{\alpha}_c$. Both results are summarized in the following theorem.

Theorem 3. *The calibration estimator $\hat{\alpha}_c$ is an unbiased estimator for α:*

$$B\left[\hat{\alpha}_c\right] = 0. \tag{16}$$

The variance of $\hat{\alpha}_c$ is equal to the following expression:

$$\begin{aligned} V(\hat{\alpha}_c) = &\left[\frac{(1-\alpha)(1-p_{00}) + \alpha p_{11}}{n} + \frac{(1-\alpha)p_{00} + \alpha(1-p_{11})}{n^2}\right] \\ &\times \left[\frac{\alpha p_{11}}{(1-\alpha)(1-p_{00}) + \alpha p_{11}}\left(1 - \frac{\alpha p_{11}}{(1-\alpha)(1-p_{00}) + \alpha p_{11}}\right)\right] \\ &+ \left[\frac{(1-\alpha)p_{00} + \alpha(1-p_{11})}{n} + \frac{(1-\alpha)(1-p_{00}) + \alpha p_{11}}{n^2}\right] \\ &\times \left[\frac{(1-\alpha)p_{00}}{(1-\alpha)p_{00} + \alpha(1-p_{11})}\left(1 - \frac{(1-\alpha)p_{00}}{(1-\alpha)p_{00} + \alpha(1-p_{11})}\right)\right] \\ &+ O\left(\max\left[\frac{1}{n^3}, \frac{1}{Nn}\right]\right). \end{aligned} \tag{17}$$

Proof. See the Appendix.

Hereby, the overview of the five estimators for α is complete. The expressions that we have derived for the bias and variance of these five estimators will now be used to compare the (root) mean squared error of the five estimators, both theoretically as well as by means of simulation studies.

3 Results

The aim of this section is to derive empirically which of the five estimators of α that we presented in Section 2 has the lowest MSE, and under which conditions. For a given population size N, the MSE of each estimator depends on four parameters (i.e, $\alpha, p_{00}, p_{11}, n$), so visualizations would have to be 5-dimensional. To reduce dimensions, we will first present a simulation study in which all four parameters are fixed. For the fixed parameter setting, the sampling distributions of the estimators are compared using boxplots. Second, we will fix several values of α and n and use plots to compare the MSE of the estimators for varying p_{00} and p_{11}. The latter analysis will already be sufficient in order to reach a final conclusion on which estimator has the lowest MSE.[‖]

3.1 Sampling distributions of the estimators

Here, we present two simple simulation studies to gain some intuition for the difference in the sampling distributions of the five estimators. In the first simulation study, we consider a class-balanced dataset, that is, $\alpha = 0.5$, with a small test dataset of size $n = 1000$, a large population dataset $N = 3 \times 10^5$ and a rather poor classifier having classification probabilities $p_{00} = 0.6$ and $p_{11} = 0.7$. We deliberately choose $p_{00} \neq p_{11}$, as otherwise the classify-and-count estimator $\hat{\alpha}^*$ would be unbiased: (p_{00}, p_{11}) would be on the line between $(1-\alpha, \alpha)$ and $(1, 1)$, see also Equation (5).

Table 2 summarizes the bias, variance and root mean squared error (RMSE), computed using the analytic approximations presented in Section 2. The classify-and-count estimator is highly biased and therefore it has a high RMSE, despite having the lowest variance of all estimators. The RMSE of the classify-and-count estimator can indeed be improved by subtracting an estimate of the bias ($\hat{\alpha}_b$). The subtraction reduces the absolute bias and only slightly increases the variance. A further bias reduction is obtained by the misclassification estimator $\hat{\alpha}_p$. However, inverting the row-normalized confusion matrix $\mathbf{P}$ (that is, the misclassification probabilities) for values of p_{00} and p_{11} close to $p_{00}+p_{11} = 1$ significantly increases the variance of the estimator, leading to the highest RMSE of all estimators considered. Finally, the calibration estimator $\hat{\alpha}_c$ is unbiased and has

[‖] The results in this section have been obtained using the statistical software R. All visualizations have been implemented in a Shiny dashboard, which in addition includes interactive 3D-plots of the RMSE surface for each of the estimators. The code can be retrieved from `https://github.com/kevinkloos/Misclassification-Bias`.

the lowest variance among the estimators that make use of the test dataset. In particular, note that the variance is also lower than that of the baseline estimator. In this example, the estimator based on the calibration probabilities has the lowest RMSE, and it is the only estimator with a lower RMSE than the baseline estimator $\hat{\alpha}_a$.

Table 2: A comparison of the bias, variance and RMSE of each of the five estimators for α, where $\alpha = 0.5$, $p_{00} = 0.6$, $p_{11} = 0.7$, $n = 1000$ and $N = 3 \times 10^5$.

Estimator	*Symbol*	Bias $\times 10^{-2}$	Variance $\times 10^{-4}$	RMSE $\times 10^{-2}$
Baseline	$\hat{\alpha}_a$	0.000	2.500	1.581
Classify-and-count	$\hat{\alpha}^*$	5.000	0.000	5.000
Subtracted-bias	$\hat{\alpha}_b$	-3.500	2.244	3.807
Misclassification	$\hat{\alpha}_p$	-0.033	25.025	5.003
Calibration	$\hat{\alpha}_c$	0.000	2.275	1.508

To gain insight in the sampling distribution of the estimators, in addition to the metrics presented in Table 2, we simulated a large number $R = 10000$ of confusion matrices for datasets of size $n = 1000$ and $N = 3 \times 10^5$. Each confusion matrix was created as follows. First, take a random draw from a $Bin(N, \alpha)$-distribution, resulting in a number N_{1+}. Then, take a random draw from a $Bin(N_{1+}, p_{11})$-distribution to obtain N_{11} and a random draw from a $Bin(N - N_{+1}, p_{00})$-distribution to obtain N_{00}. This computes the theoretical confusion matrix for the target population. Use this confusion matrix to draw a sample from a multivariate hypergeometric distribution, with its parameters from the drawn theoretical confusion matrix. These draws precisely give the number of true and false positives and negatives needed to fill a confusion matrix. Each confusion matrix can be used to compute the five estimators. Repeating this procedure $R = 10000$ times gave rise to the sampling distributions of the five estimators as presented in Figure 1. It nicely visualizes the bias and variance of the five estimators, supporting the results in Table 2. In addition, it shows that, due to the bias, the variances of the classify-and-count estimator $\hat{\alpha}^*$ and the subtracted-bias estimator $\hat{\alpha}_b$ cannot be used to obtain reliable confidence intervals for α.

In the second simulation study, we consider a highly imbalanced dataset, namely $\alpha = 0.98$. We again assume that the available test dataset has size $n = 1000$, but we assume a classifier having classification probabilities $p_{00} = 0.94$ and $p_{11} = 0.97$. Table 3 summarizes the bias, variance and RMSE of each of the estimators and Figure 2 shows the sampling distributions of each of the estimators. It can be noticed that subtracted-bias estimator and the misclassification estimator both have estimates of α that exceed 1. It is obvious that such values

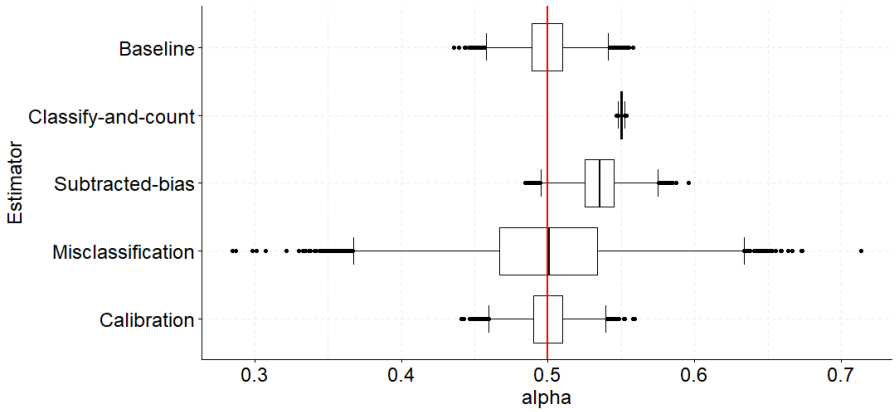


Fig. 1: The boxplots show the sampling distribution of the estimators for α, where $\alpha = 0.5$, $p_{00} = 0.6$, $p_{11} = 0.7$, $n = 1000$ and $N = 3 \times 10^5$. The true value of α is highlighted by a vertical line.

cannot occur in the population. For the method with the misclassification probabilities, this effect gets stronger when $p_{00} + p_{11}$ gets closer to 1. Furthermore, the baseline estimator performs well compared to the other estimators when the dataset is highly imbalanced: its RMSE is slightly higher than the RMSE of the method with calibration probabilities and much lower than the method with the misclassification probabilities. Finally, it is shown that the classify-and-count estimator is highly biased, even though p_{00} and p_{11} are both fairly close to 1.

Table 3: A comparison of the bias, variance and RMSE of each of the five estimators for α, where $\alpha = 0.98$, $p_{00} = 0.94$, $p_{11} = 0.97$, $n = 1000$ and $N = 3 \times 10^5$.

Method	*Symbol*	Bias $\times 10^{-2}$	Variance $\times 10^{-5}$	RMSE $\times 10^{-3}$
Baseline	$\hat{\alpha}_a$	0.000	1.960	4.427
Classify-and-count	$\hat{\alpha}^*$	-2.820	0.000	28.200
Subtracted-bias	$\hat{\alpha}_b$	0.254	3.377	6.342
Misclassification	$\hat{\alpha}_p$	-0.003	3.587	5.989
Calibration	$\hat{\alpha}_c$	0.000	1.289	3.591

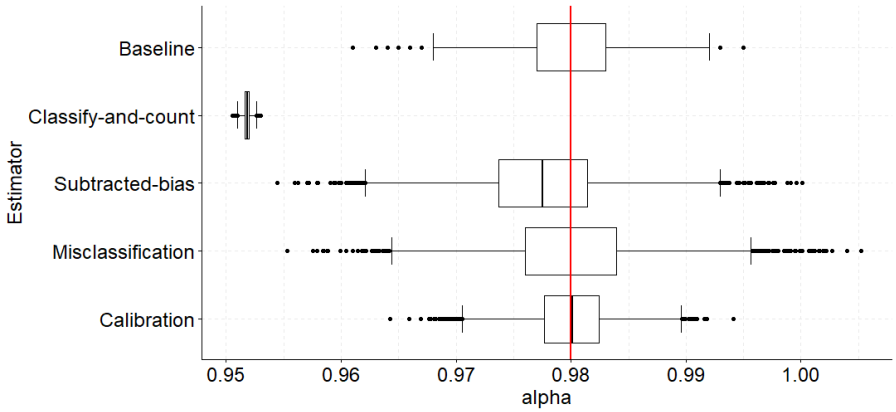


Fig. 2: The boxplots show the sampling distribution of the estimators for α, where $\alpha = 0.98$, $p_{00} = 0.94$, $p_{11} = 0.97$, $n = 1000$ and $N = 3 \times 10^5$. The true value of α is highlighted by a vertical line.

3.2 Finding the optimal estimator

The aim of this subsection is to find the optimal estimator, i.e., the estimator with the lowest RMSE, for every combination of values of the parameters α, p_{00}, p_{11} and n. First, suppose that (p_{00}, p_{11}) is close to the line in the plane through the points $(1 - \alpha, \alpha)$ and $(1, 1)$. As noted before, it implies that the classify-and-count estimator $\hat{\alpha}^*$ has low bias. Consequently, the subtracted-bias estimator $\hat{\alpha}_b$ has low bias as well. Thus, these two estimators will have the lowest RMSE in the described region, whose size decreases as n increases. Figure 3 visualizes the described region for $\alpha = 0.2$ and two different values of n. We remark that the biased estimators $\hat{\alpha}^*$ and $\hat{\alpha}_b$ perform worse (relative to the other estimators) when the sample size n of the test dataset increases. The biased methods, like Classify-and-count and Subtracted-bias, perform well when the classification probabilities are high for the largest group.

As we have seen in both Table 2 and Table 3, the calibration estimator $\hat{\alpha}_c$ competes with the baseline estimator in having the lowest RMSE. In general, the calibration estimator will have lower RMSE if the classification probabilities p_{00} and p_{11} are higher, while the baseline estimator does not depend on these classification probabilities. In a neighbourhood of $p_{00} = p_{11} = 0.5$, the baseline estimator will always have lower RMSE than the calibration estimator. However, for every α and n, there must exist a curve in the (p_{00}, p_{11})-plane beyond which the calibration estimator will have lower RMSE than the baseline estimator. The left-hand panels in Figure 4 show this curve for $\alpha = 0.2$ and two different values of n. For larger values of n, the curve where the calibration estimator performs better than the baseline estimator gets closer to $p_{00} = p_{11} = 0.5$ and therefore covers a larger area in the (p_{00}, p_{11})-plane.

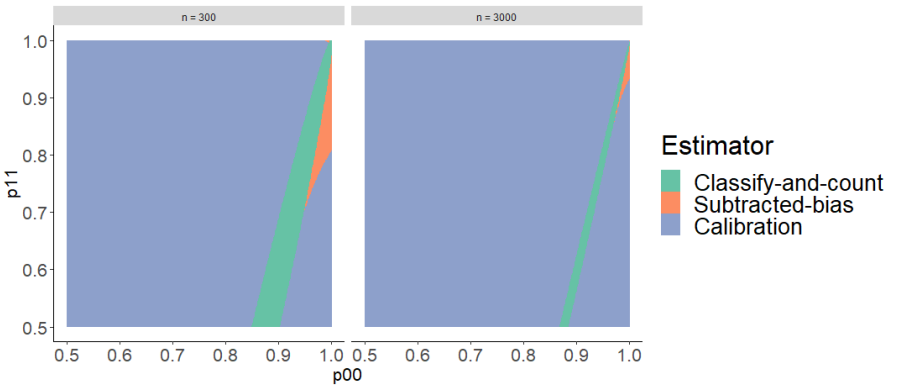


Fig. 3: For each coordinate (p_{00}, p_{11}), the depicted color indicates which estimator has the lowest RMSE, considering only the classify-and-count estimator (green), the subtracted-bias estimator (orange) and the calibration estimator (purple). In the left panel , we have set $\alpha = 0.2$ and $n = 300$, whereas $\alpha = 0.2$ and $n = 3000$ in the right panel. The red and green regions are smaller in the right panel, as the variance of the calibration estimator is decreasing in n, while the bias of the classify-and-count estimator and of the subtracted-bias estimator do not depend on n.

Table 2 and Table 3 have shown that the misclassification estimator only performs well if p_{00} and p_{11} are high, which is confirmed by the expression of the bias and variance: both have a singularity at $p_{00} + p_{11} = 1$, see Equations (13) and (14). The right-hand panels in Figure 4 show, for $\alpha = 0.2$ and two different values of n, the curve in the (p_{00}, p_{11})-plane beyond which the misclassification estimator has lower RMSE than the baseline estimator. Observe that an increase in the size n of the test dataset does not have much impact on the position of the curve. The reason is that the misclassification estimator has a singularity at $p_{00} = p_{11} = 0.5$. The shape of the curve also depends on the value of α. If $\alpha = 0.8$ instead of 0.2, the curves are line-symmetric in the line $p_{00} = p_{11}$. The curve is also line symmetric in $p_{00} = p_{11}$ for $\alpha = 0.5$. The area where the misclassification estimator performs better than the baseline estimator decreases when α gets closer towards 0 or 1. The main reason why this happens is that the variance of the baseline estimator decreases fast when α gets closer towards 0 or 1. Thus, the baseline estimator performs better than the misclassification estimator either if the classifier performs badly in general or performs badly in classifying the largest group.

The final analysis of this paper is to compare the calibration estimator and the misclassification estimator for high values of p_{00} and p_{11}. In Theorem 4 it is proven that, for all possible combinations of α and sufficiently large n, the MSE of the calibration estimator is consistently lower than that of the misclassification estimator.

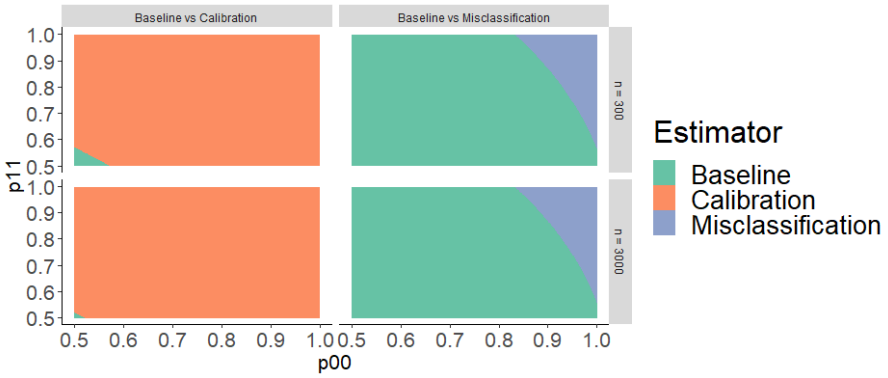


Fig. 4: For each coordinate (p_{00}, p_{11}), the depicted color indicates which estimate has the lowest RMSE, considering only the baseline estimator (green), the calibration estimator (orange) and the misclassification estimator (purple). The top-row panels consider $\alpha = 0.2$ and $n = 300$, while the bottom-row panels consider $\alpha = 0.2$ and $n = 3000$.

Theorem 4. *Let* $\widetilde{MSE}[\hat{\alpha}_p]$ *and* $\widetilde{MSE}[\hat{\alpha}_c]$ *denote the approximate mean squared errors, up to terms of order* $1/n$*, of the misclassification estimator and the calibration estimator, respectively. It holds that:*

$$\widetilde{MSE}[\hat{\alpha}_p] - \widetilde{MSE}[\hat{\alpha}_c] = \frac{\Big[(1-\alpha)p_{00}(1-p_{00}) + \alpha p_{11}(1-p_{11})\Big]^2}{(p_{00}+p_{11}-1)^2\beta(1-\beta)}, \tag{18}$$

in which $\beta := (1-\alpha)(1-p_{00}) + \alpha p_{11}$.

Proof. See the Appendix.

Thus, neglecting terms of order $1/n^2$ and higher, the result implies that the calibration estimator has a lower mean squared error than the misclassification estimator, except that both are equal if and only if $p_{00} = p_{11} = 1$. (Note that $0 < \beta < 1$.)

We do remark that the difference in MSE is large in particular for values of p_{00} and p_{11} close to $\frac{1}{2}$. More specifically, it diverges when $p_{00} + p_{11} \to 1$. It is the result of the misclassification estimator having a singularity at $p_{00} + p_{11} = 1$ (see Equation (14)), while the variance of the calibration estimator is bounded. An unpleasant consequence of the singularity at $p_{00} + p_{11} = 1$ is that, for fixed n and α, the probability that $\hat{\alpha}_p$ takes values outside the interval $[0, 1]$ increases as $p_{00} + p_{11} \to 1$; see [14] for a discussion and a possible solution.

4 Conclusion and Discussion

In this paper, we have studied the effect of classification errors on five estimators of the base rate parameter α that are obtained from machine learning algorithms.

In general, a straightforward classify-and-count estimator will lead to biased estimates and some form of bias correction should be considered. As reducing bias might increase variance, we evaluated the (root) mean squared error (MSE) of the five estimators, both theoretically as well as numerically.

From our results we may draw the following main (three-part) conclusion regarding which estimator for α has lowest mean squared error. First, when dealing with small test datasets and rather poor algorithms, that is p_{00} and p_{11} both close to 0.5, the baseline estimator $\hat{\alpha}_a$ has the lowest MSE. Second, when dealing with algorithms for which the classification probabilities p_{00} and p_{11} are in a small neighbourhood around the line $(p_{11} - 1)\alpha + (1 - p_{00})(1 - \alpha) = 0$ in the (p_{00}, p_{11})-plane, the classify-and-count estimator and the subtracted-bias estimator will have the lowest MSE. As the size of the test dataset increases, the size of that neighbourhood decreases. Third, in any other situation, the calibration estimator will have the lowest MSE. In practice, the test dataset will have to be used to determine which of the three scenarios applies to the data and the algorithm at hand. It is an additional estimation problem that we have not discussed in this paper.

We would like to close the paper by pointing out three interesting directions for future research. First, the results could be generalized to multi-class classification problems. The theoretical derivations of the bias and variance are more complicated and involve matrix-vector notation, but the proof strategy is similar. However, it is more challenging to compare the MSE of the five estimators visually in the multi-class case.

Second, the assumptions that we have made could be relaxed. In particular, a trained and implemented machine learning model is, in practice, often used over a longer period of time. A shift in the base rate parameter α, also known as prior probability shift [15], is then inevitable. Consequently, we may no longer assume that the conditional distribution of the class label given the features in the test dataset is similar to that in the population. It implies that the calibration estimator is no longer unbiased, which might have a significant effect on our main conclusion.

Third and finally, a combination of estimators might have a substantially lower MSE than that of the individual estimators separately. Therefore, it might be interesting to study different methods of model averaging applied to the problem of misclassification bias. It could be fruitful especially when the assumptions that we have made are relaxed.

References

1. Buonaccorsi, J.P.: Measurement Error: Models, Methods, and Applications. Chapman & Hall/CRC, Boca Raton, FL (2010)
2. Burger, J., Delden, A.v., Scholtus, S.: Sensitivity of Mixed-Source Statistics to Classification Errors. Journal of Official Statistics **31**(3), 489–506 (2015)
3. Curier, R., De Jong, T., Strauch, K., Cramer, K., Rosenski, N., Schartner, C., Debusschere, M., Ziemons, H., Iren, D., Bromuri, S.: Monitoring spatial sustainable

development: Semi-automated analysis of satellite and aerial images for energy transition and sustainability indicators. arXiv preprint arXiv:1810.04881 (2018)
4. Czaplewski, R.L.: Misclassification bias in areal estimates. Photogrammetric Engineering and Remote Sensing. 58(2): 189-192 **58**(2), 189–192 (1992)
5. Czaplewski, R.L., Catts, G.P.: Calibration of remotely sensed proportion or area estimates for misclassification error. Remote Sensing of Environment **39**(1), 29–43 (1992)
6. González, P., Castaño, A., Chawla, N.V., Coz, J.J.D.: A Review on Quantification Learning. ACM Computing Surveys **50**(5), 74:1–74:40 (2017). https://doi.org/10.1145/3117807
7. Grassia, A., Sundberg, R.: Statistical Precision in the Calibration and Use of Sorting Machines and Other Classifiers. Technometrics **24**(2), 117–121 (1982)
8. Greenland, S.: Sensitivity Analysis and Bias Analysis. In: Ahrens, W., Pigeot, I. (eds.) Handbook of Epidemiology. Springer, New York, NY (2014)
9. Hopkins, D.J., King, G.: A Method of Automated Nonparametric Content Analysis for Social Science. American Journal of Political Science **54**(1), 229–247 (2010)
10. Knottnerus, P.: Sample survey theory: some Pythagorean perspectives. Springer Science & Business Media (2003)
11. Kuha, J., Skinner, C.J.: Categorical data analysis and misclassification. In: Lyberg, L., Biemer, P., Collins, M., de Leeuw, E., Dippo, C., Schwarz, N., Trewin, D. (eds.) Survey Measurement and Process Quality, pp. 633–670. Wiley (Mar 1997)
12. Löw, F., Knöfel, P., Conrad, C.: Analysis of uncertainty in multi-temporal object-based classification. ISPRS Journal of Photogrammetry and Remote Sensing **105**, 91–106 (2015)
13. Meertens, Q.A., Diks, C.G.H., Herik, H.J.v.d., Takes, F.W.: A data-driven supply-side approach for estimating cross-border Internet purchases within the European Union. Journal of the Royal Statistical Society: Series A (Statistics in Society) **183**(1), 61–90 (2020). https://doi.org/10.1111/rssa.12487
14. Meertens, Q., Diks, C., van den Herik, H., Takes, F.: A Bayesian Approach for Accurate Classification-Based Aggregates. Society for Industrial and Applied Mathematics, Philadelphia, PA (2019)
15. Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012). https://doi.org/10.1016/j.patcog.2011.06.019
16. O'Connor, B., Balasubramanyan, R., Routledge, B., Smith, N.: From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In: Proceedings of the International AAAI Conference on Weblogs and Social Media. Washington, DC (2010)
17. Scholtus, S., van Delden, A.: On the accuracy of estimators based on a binary classifier (Feb 2020), Discussion Paper, Statistics Netherlands, The Hague
18. Schwartz, J.E.: The Neglected Problem of Measurement Error in Categorical Data. Sociological Methods & Research **13**(4), 435–466 (1985). https://doi.org/10.1177/0049124185013004001
19. Strichartz, R.S.: The Way of Analysis. Jones & Bartlett Learning (2000)
20. Van Delden, A., Scholtus, S., Burger, J.: Accuracy of mixed-source statistics as affected by classification errors. Journal of Official Statistics **32**(3), 619–642 (2016)
21. Wiedemann, G.: Proportional Classification Revisited: Automatic Content Analysis of Political Manifestos Using Active Learning. Social Science Computer Review **37**(2), 135–159 (2019)

Appendix

This appendix contains the proofs of the theorems presented in the paper entitled "Comparing Correction Methods for Misclassification Bias". Recall that we have assumed a population of size N in which a fraction $\alpha \coloneqq N_{1+}/N$ belongs to the class of interest, referred to as the class labelled as 1. We assume that a binary classification algorithm has been trained that correctly classifies a data point that belongs to class $i \in \{0, 1\}$ with probability $p_{ii} > 0.5$, independently across all data points. In addition, we assume that a test set of size $n \ll N$ is available and that it can be considered a simple random sample from the population. The classification probabilities p_{00} and p_{11} are estimated on that test set as described in Section 2. Finally, we assume that the classify-and-count estimator $\hat{\alpha}^*$ is distributed independently of $\hat{p}_{00}$ and $\hat{p}_{11}$, which is reasonable (at least as an approximation) when $n \ll N$.

It may be noted that the estimated probabilities $\hat{p}_{11}$ and $\hat{p}_{00}$ defined in Section 2 cannot be computed if $n_{1+} = 0$ or $n_{0+} = 0$. Similarly, the calibration probabilities c_{11} and c_{00} cannot be estimated if $n_{+1} = 0$ or $n_{+0} = 0$. We assume here that these events occur with negligible probability. This will be true when n is sufficiently large so that $n\alpha \gg 1$ and $n(1-\alpha) \gg 1$.

Preliminaries

Many of the proofs presented in this appendix rely on the following two mathematical results. First, we will use univariate and bivariate Taylor series to approximate the expectation of non-linear functions of random variables. That is, to estimate $E[f(X)]$ and $E[g(X, Y)]$ for sufficiently differentiable functions f and g, we will insert the Taylor series for f and g at $x_0 = E[X]$ and $y_0 = E[Y]$ up to terms of order 2 and utilize the linearity of the expectation. Second, we will use the following conditional variance decomposition for the variance of a random variable X:

$$V(X) = E[V(X \mid Y)] + V(E[X \mid Y]). \tag{19}$$

The conditional variance decomposition follows from the tower property of conditional expectations [10]. Before we prove the theorems presented in the paper, we begin by proving the following lemma.

Lemma 1. *The variance of the estimator $\hat{p}_{11}$ for p_{11} estimated on the test set is given by*

$$V(\hat{p}_{11}) = \frac{p_{11}(1-p_{11})}{n\alpha}\left[1 + \frac{1-\alpha}{n\alpha}\right] + O\left(\frac{1}{n^3}\right). \tag{20}$$

Similarly, the variance of $\hat{p}_{00}$ is given by

$$V(\hat{p}_{00}) = \frac{p_{00}(1-p_{00})}{n(1-\alpha)}\left[1 + \frac{\alpha}{n(1-\alpha)}\right] + O\left(\frac{1}{n^3}\right). \tag{21}$$

Moreover, $\hat{p}_{11}$ and $\hat{p}_{00}$ are uncorrelated: $C(\hat{p}_{11}, \hat{p}_{00}) = 0$.

Proof (of Lemma 1). We approximate the variance of $\hat{p}_{00}$ using the conditional variance decomposition and a second-order Taylor series, as follows:

$$\begin{aligned}
V(\hat{p}_{00}) &= V\left(\frac{n_{00}}{n_{0+}}\right) \\
&= E_{n_{0+}}\left[V\left(\frac{n_{00}}{n_{0+}} \mid n_{0+}\right)\right] + V_{n_{0+}}\left[E\left(\frac{n_{00}}{n_{0+}} \mid n_{0+}\right)\right] \\
&= E_{n_{0+}}\left[\frac{1}{n_{0+}^2}V(n_{00} \mid n_{0+})\right] + V_{n_{0+}}\left[\frac{1}{n_{0+}}E(n_{00} \mid n_{0+})\right] \\
&= E_{n_{0+}}\left[\frac{n_{0+}p_{00}(1-p_{00})}{n_{0+}^2}\right] + V_{n_{0+}}\left[\frac{n_{0+}p_{00}}{n_{0+}}\right] \\
&= E_{n_{0+}}\left[\frac{1}{n_{0+}}\right]p_{00}(1-p_{00}) \\
&= \left[\frac{1}{E[n_{0+}]} + \frac{1}{2}\frac{2}{E[n_{0+}]^3} \times V[n_{0+}]\right]p_{00}(1-p_{00}) + O\left(\frac{1}{n^3}\right) \\
&= \frac{p_{00}(1-p_{00})}{E[n_{0+}]}\left[1 + \frac{V[n_{0+}]}{E[n_{0+}]^2}\right] + O\left(\frac{1}{n^3}\right) \\
&= \frac{p_{00}(1-p_{00})}{n(1-\alpha)}\left[1 + \frac{\alpha}{n(1-\alpha)}\right] + O\left(\frac{1}{n^3}\right).
\end{aligned}$$

The variance of $\hat{p}_{11}$ is approximated in the exact same way.

Finally, to evaluate $C(\hat{p}_{11}, \hat{p}_{00})$ we use the analogue of (19) for covariances:

$$\begin{aligned}
C(\hat{p}_{11}, \hat{p}_{00}) &= C\left(\frac{n_{11}}{n_{1+}}, \frac{n_{00}}{n_{0+}}\right) \\
&= E_{n_{1+},n_{0+}}\left[C\left(\frac{n_{11}}{n_{1+}}, \frac{n_{00}}{n_{0+}} \mid n_{1+}, n_{0+}\right)\right] \\
&\quad + C_{n_{1+},n_{0+}}\left[E\left(\frac{n_{11}}{n_{1+}} \mid n_{1+}, n_{0+}\right), E\left(\frac{n_{00}}{n_{0+}} \mid n_{1+}, n_{0+}\right)\right] \\
&= E_{n_{1+},n_{0+}}\left[\frac{1}{n_{1+}n_{0+}}C(n_{11}, n_{00} \mid n_{1+}, n_{0+})\right] \\
&\quad + C_{n_{1+},n_{0+}}\left[\frac{1}{n_{1+}}E(n_{11} \mid n_{1+}), \frac{1}{n_{0+}}E(n_{00} \mid n_{0+})\right].
\end{aligned}$$

The second term is zero as before. The first term also vanishes because, conditional on the row totals n_{1+} and n_{0+}, the counts n_{11} and n_{00} follow independent binomial distributions, so $C(n_{11}, n_{00} \mid n_{1+}, n_{0+}) = 0$.

Note: in the remainder of this appendix, we will not add explicit subscripts to expectations and variances when their meaning is unambiguous.

Subtracted-bias estimator

We will now prove the bias and variance approximations for the subtracted-bias estimator $\hat{\alpha}_b$ that was defined in Equation (9).

Proof (of Theorem 1). The bias of $\hat{\alpha}_b$ is given by

$$\begin{aligned} B(\hat{\alpha}_b) &= E\left[\hat{\alpha}^\star - \hat{B}[\hat{\alpha}^\star]\right] - \alpha \\ &= E[\hat{\alpha}^\star - \alpha] - E\left[\hat{B}[\hat{\alpha}^\star]\right] \\ &= B[\hat{\alpha}^\star] - E\left[\hat{B}[\hat{\alpha}^\star]\right] \\ &= [\alpha(p_{00} + p_{11} - 2) + (1 - p_{00})] - E\left[\hat{\alpha}^\star(\hat{p}_{00} + \hat{p}_{11} - 2) + (1 - \hat{p}_{00})\right]. \end{aligned}$$

Because $\hat{\alpha}^*$ and $(\hat{p}_{00} + \hat{p}_{11} - 2)$ are assumed to be independent, the expectation of their product equals the product of their expectations:

$$\begin{aligned} B(\hat{\alpha}_b) &= \alpha(p_{00} + p_{11} - 2) + (1 - p_{00}) - E[\hat{\alpha}^\star](p_{00} + p_{11} - 2) - (1 - p_{00}) \\ &= (\alpha - E[\hat{\alpha}^\star])(p_{00} + p_{11} - 2) \\ &= B[\hat{\alpha}^\star](2 - p_{00} - p_{11}) \\ &= (1 - p_{00})(2 - p_{00} - p_{11}) - \alpha(p_{00} + p_{11} - 2)^2. \end{aligned}$$

This proves the formula for the bias of $\hat{\alpha}_b$ as estimator for α. To approximate the variance of $\hat{\alpha}_b$, we apply the conditional variance decomposition (19) conditional on $\hat{\alpha}^*$ and look at the two resulting terms separately. First, consider the expectation of the conditional variance:

$$\begin{aligned} E\left[V(\hat{\alpha}_b \mid \hat{\alpha}^*)\right] &= E\left[V(\hat{\alpha}^*(3 - \hat{p}_{00} - \hat{p}_{11}) - (1 - \hat{p}_{00}) \mid \hat{\alpha}^*)\right] \\ &= E\big[V(\hat{\alpha}^*(3 - \hat{p}_{00} - \hat{p}_{11}) \mid \hat{\alpha}^*) + V(1 - \hat{p}_{00} \mid \hat{\alpha}^*) \\ &\qquad - 2C(\hat{\alpha}^*(3 - \hat{p}_{00} - \hat{p}_{11}), 1 - \hat{p}_{00} \mid \hat{\alpha}^*)\big] \\ &= E\big[(\hat{\alpha}^*)^2 V(3 - \hat{p}_{00} - \hat{p}_{11} \mid \hat{\alpha}^*) + V(1 - \hat{p}_{00} \mid \hat{\alpha}^*) \\ &\qquad - 2\hat{\alpha}^* C(3 - \hat{p}_{00} - \hat{p}_{11}, 1 - \hat{p}_{00} \mid \hat{\alpha}^*)\big] \\ &= E\big[(\hat{\alpha}^*)^2 \left[V(\hat{p}_{00}) + V(\hat{p}_{11})\right] + V(\hat{p}_{00}) - 2\hat{\alpha}^* V(\hat{p}_{00})\big] \\ &= E\left[(\hat{\alpha}^*)^2\right] \left[V(\hat{p}_{00}) + V(\hat{p}_{11})\right] + V(\hat{p}_{00}) - 2E\left[\hat{\alpha}^*\right] V(\hat{p}_{00}). \end{aligned}$$

In the penultimate line, we used that $C(\hat{p}_{11}, \hat{p}_{00}) = 0$. The second moment $E\left[(\hat{\alpha}^*)^2\right]$ can be written as $E\left[\hat{\alpha}^*\right]^2 + V(\hat{\alpha}^*)$. Because $V(\hat{\alpha}^*)$ is of order $1/N$, it can be neglected compared to $E\left[\hat{\alpha}^*\right]^2$, which is of order 1. In particular, we find that the expectation of the conditional variance equals:

$$\begin{aligned} E\left[V(\hat{\alpha}_b \mid \hat{\alpha}^*)\right] &= E\left[(\hat{\alpha}^*)\right]^2 \left[V(\hat{p}_{00}) + V(\hat{p}_{11})\right] + V(\hat{p}_{00}) - 2E\left[\hat{\alpha}^*\right] V(\hat{p}_{00}) + O\left(\frac{1}{N}\right) \\ &= V(\hat{p}_{00}) \left[E\left[\hat{\alpha}^*\right] - 1\right]^2 + V(\hat{p}_{11}) E\left[\hat{\alpha}^*\right]^2 + O\left(\frac{1}{N}\right). \end{aligned}$$

Next, the variance of the conditional expectation can be seen to be equal the following:

$$\begin{aligned} V\left[E(\hat{\alpha}_b \mid \hat{\alpha}^*)\right] &= V\left[E(\hat{\alpha}^*(3 - \hat{p}_{00} - \hat{p}_{11}) - (1 - \hat{p}_{00}) \mid \hat{\alpha}^*)\right] \\ &= V\left[\hat{\alpha}^* E(3 - \hat{p}_{00} - \hat{p}_{11} \mid \hat{\alpha}^*) - E(1 - \hat{p}_{00} \mid \hat{\alpha}^*)\right] \\ &= V(\hat{\alpha}^*)(3 - p_{00} - p_{11})^2. \end{aligned}$$

Because $V(\hat{\alpha}^*)$ is of order $1/N$, it can be neglected in the final formula. Furthermore, the variances of $\hat{p}_{00}$ and $\hat{p}_{11}$ can be written out using the result from Lemma 1:

$$\begin{aligned} V(\hat{\alpha}_b) = & \frac{\left[\alpha(p_{00}+p_{11}-1)-p_{00}\right]^2 p_{00}(1-p_{00})}{n(1-\alpha)}\left[1+\frac{\alpha}{n(1-\alpha)}\right] \\ & + \frac{\left[\alpha(p_{00}+p_{11}-1)+(1-p_{00})\right]^2 p_{11}(1-p_{11})}{n\alpha}\left[1+\frac{1-\alpha}{n\alpha}\right] \\ & + O\left(\max\left[\frac{1}{n^3},\frac{1}{N}\right]\right). \end{aligned}$$

This concludes the proof of Theorem 1.

Misclassification estimator

We will now prove the bias and variance approximations for the misclassification estimator $\hat{\alpha}_p$ as defined in Equation (12).

Proof (of Theorem 2). Under the assumption that $\hat{\alpha}^*$ is distributed independently of $(\hat{p}_{00}, \hat{p}_{11})$, it holds that

$$\begin{aligned} E(\hat{\alpha}_p) &= E\left(\frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right) + E\left[E\left(\frac{\hat{\alpha}^*}{\hat{p}_{00}+\hat{p}_{11}-1}\;\middle|\;\hat{\alpha}^*\right)\right] \\ &= E\left(\frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right) + E(\hat{\alpha}^*)E\left(\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}\right). \end{aligned} \tag{22}$$

$E(\hat{\alpha}^*)$ is known from (4). To evaluate the other two expectations, we use a second-order Taylor series approximation. The first- and second-order partial derivatives of $f(x,y) = 1/(x+y-1)$ and $g(x,y) = (x-1)/(x+y-1) = 1-[y/(x+y-1)]$ are given by:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = \frac{-1}{(x+y-1)^2}, \tag{23}$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial y^2} = \frac{2}{(x+y-1)^3},$$

$$\frac{\partial g}{\partial x} = \frac{y}{(x+y-1)^2}, \tag{24}$$

$$\frac{\partial g}{\partial y} = \frac{-(x-1)}{(x+y-1)^2}, \tag{25}$$

$$\frac{\partial^2 g}{\partial x^2} = \frac{-2y}{(x+y-1)^3},$$

$$\frac{\partial^2 g}{\partial y^2} = \frac{2(x-1)}{(x+y-1)^3}.$$

Now also using that $C(\hat{p}_{11}, \hat{p}_{00}) = 0$, we obtain for the first expectation:

$$E\left(\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}\right) = \frac{1}{p_{00}+p_{11}-1} + \frac{V(\hat{p}_{00})+V(\hat{p}_{11})}{(p_{00}+p_{11}-1)^3} + O(n^{-2})$$
$$= \frac{1}{p_{00}+p_{11}-1}\left[1 + \frac{\frac{p_{00}(1-p_{00})}{n(1-\alpha)} + \frac{p_{11}(1-p_{11})}{n\alpha}}{(p_{00}+p_{11}-1)^2}\right] + O(n^{-2}). \tag{26}$$

Here, we have included only the first term of the approximations to $V(\hat{p}_{00})$ and $V(\hat{p}_{11})$ from Lemma 1, since this suffices to approximate the bias up to terms of order $O(1/n)$. Similarly, for the second expectation we obtain:

$$E\left(\frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right) = \frac{p_{00}-1}{p_{00}+p_{11}-1} + \frac{(p_{00}-1)V(\hat{p}_{11}) - p_{11}V(\hat{p}_{00})}{(p_{00}+p_{11}-1)^3} + O(n^{-2})$$
$$= \frac{p_{00}-1}{p_{00}+p_{11}-1}\left[1 + p_{11}\frac{\frac{1-p_{11}}{n\alpha} + \frac{p_{00}}{n(1-\alpha)}}{(p_{00}+p_{11}-1)^2}\right] + O(n^{-2}). \tag{27}$$

Using (22), (4), (26), and (27), we conclude that:

$$E(\hat{\alpha}_p) = \frac{\alpha(p_{00}+p_{11}-1)-(p_{00}-1)}{p_{00}+p_{11}-1}\left[1 + \frac{\frac{p_{00}(1-p_{00})}{n(1-\alpha)} + \frac{p_{11}(1-p_{11})}{n\alpha}}{(p_{00}+p_{11}-1)^2}\right]$$
$$+ \frac{p_{00}-1}{p_{00}+p_{11}-1}\left[1 + p_{11}\frac{\frac{1-p_{11}}{n\alpha} + \frac{p_{00}}{n(1-\alpha)}}{(p_{00}+p_{11}-1)^2}\right] + O\left(\frac{1}{n^2}\right).$$

From this, it follows that an approximation to the bias of $\hat{\alpha}_p$ that is correct up to terms of order $O(1/n)$ is given by:

$$B(\hat{\alpha}_p) = \frac{\alpha(p_{00}+p_{11}-1)-(p_{00}-1)}{n(p_{00}+p_{11}-1)^3}\left[\frac{p_{00}(1-p_{00})}{1-\alpha} + \frac{p_{11}(1-p_{11})}{\alpha}\right]$$
$$+ \frac{(p_{00}-1)p_{11}}{n(p_{00}+p_{11}-1)^3}\left[\frac{1-p_{11}}{\alpha} + \frac{p_{00}}{1-\alpha}\right] + O\left(\frac{1}{n^2}\right).$$

By expanding the products in this expression and combining similar terms, the expression can be simplified to:

$$B(\hat{\alpha}_p) = \frac{p_{11}(1-p_{11}) - p_{00}(1-p_{00})}{n(p_{00}+p_{11}-1)^2} + O\left(\frac{1}{n^2}\right).$$

Finally, using the identity $p_{11}(1-p_{11}) - p_{00}(1-p_{00}) = (p_{00}+p_{11}-1)(p_{00}-p_{11})$, we obtain the required result for $B(\hat{\alpha}_p)$.

To approximate the variance of $\hat{\alpha}_p$, we apply the conditional variance decomposition conditional on $\hat{\alpha}^*$ and look at the two resulting terms separately. First,

consider the variance of the conditional expectation:

$$\begin{aligned} V\left[E(\hat{\alpha}_p \mid \hat{\alpha}^*)\right] &= V\left[E\left(\hat{\alpha}^* \frac{1}{\hat{p}_{00}+\hat{p}_{11}-1} + \frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1} \mid \hat{\alpha}^*\right)\right] \\ &= V\left[\hat{\alpha}^* \frac{1}{p_{00}+p_{11}-1}\right] \\ &= \frac{1}{(p_{00}+p_{11}-1)^2} V\left[\hat{\alpha}^*\right] = O\left(\frac{1}{N}\right), \end{aligned} \tag{28}$$

where in the last line we used (6). Note: the factor $1/(p_{00}+p_{11}-1)^2$ can become arbitrarily large in the limit $p_{00}+p_{11} \to 1$. It will be seen below that this same factor also occurs in the lower-order terms of $V(\hat{\alpha}_p)$; hence, the relative contribution of (28) remains negligible even in the limit $p_{00}+p_{11} \to 1$.

Next, we compute the expectation of the conditional variance.

$$\begin{aligned} E\left[V(\hat{\alpha}_p \mid \hat{\alpha}^*)\right] &= E\left[V\left(\hat{\alpha}^* \frac{1}{\hat{p}_{00}+\hat{p}_{11}-1} + \frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1} \mid \hat{\alpha}^\star\right)\right] \\ &= E\Bigg[V\left(\hat{\alpha}^* \frac{1}{\hat{p}_{00}+\hat{p}_{11}-1} \mid \alpha^\star\right) + V\left(\frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1} \mid \hat{\alpha}^\star\right) \\ &\qquad + 2C\left(\hat{\alpha}^* \frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}, \frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1} \mid \hat{\alpha}^\star\right)\Bigg] \\ &= E\left[(\hat{\alpha}^*)^2\right] V\left[\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] + V\left[\frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] \\ &\qquad + 2E\left[\hat{\alpha}^\star\right] C\left[\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}, \frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] \\ &= E\left[\hat{\alpha}^\star\right]^2 \left[1+O\left(\frac{1}{N}\right)\right] V\left[\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] + V\left[\frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] \\ &\qquad + 2E\left[\hat{\alpha}^\star\right] C\left[\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}, \frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right]. \end{aligned} \tag{29}$$

To approximate the variance and covariance terms, we use a first-order Taylor series. Using the partial derivatives in (23), (24) and (25), we obtain:

$$\begin{aligned} V\left[\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] &= \frac{V(\hat{p}_{00})+V(\hat{p}_{11})}{(p_{00}+p_{11}-1)^4} + O(n^{-2}) \\ V\left[\frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] &= \frac{V(\hat{p}_{00})(p_{11})^2}{(p_{00}+p_{11}-1)^4} + \frac{V(\hat{p}_{11})(1-p_{00})^2}{(p_{00}+p_{11}-1)^4} + O(n^{-2}) \\ C\left[\frac{1}{\hat{p}_{00}+\hat{p}_{11}-1}, \frac{\hat{p}_{00}-1}{\hat{p}_{00}+\hat{p}_{11}-1}\right] &= \frac{V(\hat{p}_{00})(-p_{11})}{(p_{00}+p_{11}-1)^4} + \frac{V(\hat{p}_{11})(p_{00}-1)}{(p_{00}+p_{11}-1)^4} + O(n^{-2}). \end{aligned}$$

Substituting these terms into Formula (29) and accounting for Formula (28) yields:

$$\begin{aligned}
V(\hat{\alpha}_p) &= \frac{V(\hat{p}_{00})\left[E\left[\hat{\alpha}^\star\right]^2 - 2p_{11}E\left[\hat{\alpha}^\star\right] + p_{11}^2\right]}{(p_{00}+p_{11}-1)^4} \\
&\quad + \frac{V(\hat{p}_{11})\left[E\left[\hat{\alpha}^\star\right]^2 - 2(1-p_{00})E\left[\hat{\alpha}^\star\right] + (1-p_{00})^2\right]}{(p_{00}+p_{11}-1)^4} + O\left(\max\left[\frac{1}{n^2}, \frac{1}{N}\right]\right) \\
&= \frac{V(\hat{p}_{00})\left[E\left[\hat{\alpha}^\star\right] - p_{11}\right]^2}{(p_{00}+p_{11}-1)^4} + \frac{V(\hat{p}_{11})\left[E\left[\hat{\alpha}^\star\right] - (1-p_{00})\right]^2}{(p_{00}+p_{11}-1)^4} + O\left(\max\left[\frac{1}{n^2}, \frac{1}{N}\right]\right) \\
&= \frac{V(\hat{p}_{00})(1-\alpha)^2}{(p_{00}+p_{11}-1)^2} + \frac{V(\hat{p}_{11})\alpha^2}{(p_{00}+p_{11}-1)^2} + O\left(\max\left[\frac{1}{n^2}, \frac{1}{N}\right]\right).
\end{aligned}$$

Finally, inserting the expressions for $V(\hat{p}_{00})$ and $V(\hat{p}_{11})$ from Lemma 1 yields:

$$\begin{aligned}
V(\hat{\alpha}_p) = &\frac{\frac{p_{00}(1-p_{00})}{n(1-\alpha)}\left[1 + \frac{\alpha}{n(1-\alpha)}\right](1-\alpha)^2}{(p_{00}+p_{11}-1)^2} + \frac{\frac{p_{11}(1-p_{11})}{n\alpha}\left[1 + \frac{1-\alpha}{n\alpha}\right]\alpha^2}{(p_{00}+p_{11}-1)^2} \\
&+ O\left(\max\left[\frac{1}{n^2}, \frac{1}{N}\right]\right),
\end{aligned}$$

from which expression (14) follows. This concludes the proof of Theorem 2.

Calibration estimator

We will now prove the bias and variance approximations for the calibration estimator $\hat{\alpha}_c$ that was defined in Equation (15).

Proof (of Theorem 3). To compute the expected value of $\hat{\alpha}_c$, we first compute its expectation conditional on the 4-vector $\boldsymbol{N} = (N_{00}, N_{01}, N_{10}, N_{11})$:

$$\begin{aligned}
E(\hat{\alpha}_c \mid \boldsymbol{N}) &= E\left[\hat{\alpha}^* \frac{n_{11}}{n_{+1}} + (1-\hat{\alpha}^*)\frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}\right] \\
&= \hat{\alpha}^* E\left[\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}\right] + (1-\hat{\alpha}^*)E\left[\frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}\right] \\
&= \hat{\alpha}^* E\left[E\left(\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}, n_{+1}\right) \mid \boldsymbol{N}\right] \\
&\qquad + (1-\hat{\alpha}^*)E\left[E\left(\frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}, n_{+0}\right) \mid \boldsymbol{N}\right] \\
&= \frac{N_{+1}}{N} E\left[\frac{1}{n_{+1}} n_{+1} \frac{N_{11}}{N_{+1}} \mid \boldsymbol{N}\right] + \frac{N_{+0}}{N} E\left[\frac{1}{n_{+0}} n_{+0} \frac{N_{10}}{N_{+0}} \mid \boldsymbol{N}\right] \\
&= \frac{N_{11}}{N} + \frac{N_{10}}{N} \\
&= \frac{N_{1+}}{N} = \alpha. \qquad (30)
\end{aligned}$$

By the tower property of conditional expectations, it follows that $E[\hat{\alpha}_c] = E\left[E(\hat{\alpha}_c \mid \boldsymbol{N})\right] = \alpha$. This proves that $\hat{\alpha}_c$ is an unbiased estimator for α.

To compute the variance of $\hat{\alpha}_c$, we use the conditional variance decomposition, again conditioning on the 4-vector $\boldsymbol{N}$. We remark that N_{0+} and N_{1+} are deterministic values, but that N_{+0} and N_{+1} are random variables. As shown above in Equation (30), the conditional expectation is deterministic, hence it has no variance: $V(E[\hat{\alpha}_c \mid \boldsymbol{N}]) = 0$. The conditional variance decomposition then simplifies to the following:

$$V(\hat{\alpha}_c) = E\left[V(\hat{\alpha}_c \mid \boldsymbol{N})\right]. \tag{31}$$

The conditional variance $V(\hat{\alpha}_c \mid \boldsymbol{N})$ can be written as follows:

$$\begin{aligned} V[\hat{\alpha}_c \mid \boldsymbol{N}] &= V\left[\hat{\alpha}^* \frac{n_{11}}{n_{+1}} + (1-\hat{\alpha}^*)\frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}\right] \\ &= (\hat{\alpha}^*)^2 V\left[\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}\right] + (1-\hat{\alpha}^*)^2 V\left[\frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}\right] \\ &\quad + 2\hat{\alpha}^*(1-\hat{\alpha}^*) C\left[\frac{n_{11}}{n_{+1}}, \frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}\right]. \end{aligned} \tag{32}$$

We will consider these terms separately. First, the variance of n_{11}/n_{+1} can be computed by applying an additional conditional variance decomposition:

$$V\left[\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}\right] = V\left[E\left(\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}, n_{+1}\right) \mid \boldsymbol{N}\right] + E\left[V\left(\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}, n_{+1}\right) \mid \boldsymbol{N}\right].$$

The first term is zero, which can be shown as follows:

$$\begin{aligned} V\left[E\left(\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}, n_{+1}\right)\right] &= V\left[\frac{1}{n_{+1}} E(n_{11} \mid \boldsymbol{N}, n_{+1}) \mid \boldsymbol{N}\right] \\ &= V\left[\frac{1}{n_{+1}} n_{+1} \frac{N_{11}}{N_{+1}} \mid \boldsymbol{N}\right] \\ &= V\left[\frac{N_{11}}{N_{+1}} \mid \boldsymbol{N}\right] = 0. \end{aligned}$$

For the second term, we find under the assumption that $n \ll N$:

$$\begin{aligned} E\left[V\left(\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}, n_{+1}\right) \mid \boldsymbol{N}\right] &= E\left[\frac{1}{n_{+1}^2} V(n_{11} \mid \boldsymbol{N}, n_{+1}) \mid \boldsymbol{N}\right] \\ &= E\left[\frac{1}{n_{+1}^2} n_{+1} \frac{N_{11}}{N_{+1}}\left(1 - \frac{N_{11}}{N_{+1}}\right) \mid \boldsymbol{N}\right] \\ &= E\left[\frac{1}{n_{+1}} \mid \boldsymbol{N}\right] \frac{N_{11} N_{01}}{N_{+1}^2}. \end{aligned}$$

The expectation of $\frac{1}{n_{+1}}$ can be approximated with a second-order Taylor series:

$$\begin{aligned} V\left[\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}\right] &= \left[\frac{1}{E[n_{+1} \mid \boldsymbol{N}]} + \frac{1}{2}\frac{2}{E[n_{+1} \mid \boldsymbol{N}]^3} V\left[n_{+1} \mid \boldsymbol{N}\right]\right] \frac{N_{11}N_{01}}{N_{+1}^2} + O(n^{-3}) \\ &= \frac{1}{E[n_{+1} \mid \boldsymbol{N}]}\left[1 + \frac{V\left[n_{+1} \mid \boldsymbol{N}\right]}{E[n_{+1} \mid \boldsymbol{N}]^2}\right] \frac{N_{11}N_{01}}{N_{+1}^2} + O(n^{-3}) \\ &= \frac{1}{n\hat{\alpha}^*}\left[1 + \frac{1-\hat{\alpha}^*}{n\hat{\alpha}^*}\right] \frac{N_{11}N_{01}}{N_{+1}^2} + O(n^{-3}). \end{aligned} \tag{33}$$

The variance of n_{10}/n_{+0} can be approximated in the same way, which yields the following expression:

$$V\left[\frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}\right] = \frac{1}{n(1-\hat{\alpha}^*)}\left[1 + \frac{\hat{\alpha}^*}{n(1-\hat{\alpha}^*)}\right] \frac{N_{00}N_{10}}{N_{+0}^2} + O(n^{-3}). \tag{34}$$

Finally, it can be shown that the covariance in the final term is equal to zero:

$$\begin{aligned} C\left[\frac{n_{11}}{n_{+1}}, \frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}\right] &= E\left[C\left(\frac{n_{11}}{n_{+1}}, \frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}, n_{+0}, n_{+1}\right) \mid \boldsymbol{N}\right] \\ &\quad + C\left[E\left(\frac{n_{11}}{n_{+1}} \mid \boldsymbol{N}, n_{+0}, n_{+1}\right), E\left(\frac{n_{10}}{n_{+0}} \mid \boldsymbol{N}, n_{+0}, n_{+1}\right) \mid \boldsymbol{N}\right] \\ &= E\left[\frac{1}{n_{+0}n_{+1}} C\left(n_{11}, n_{10} \mid \boldsymbol{N}, n_{+0}, n_{+1}\right) \mid \boldsymbol{N}\right] \\ &\quad + C\left[\frac{1}{n_{+1}} E\left(n_{11} \mid \boldsymbol{N}, n_{+0}, n_{+1}\right), \frac{1}{n_{+0}} E\left(n_{10} \mid \boldsymbol{N}, n_{+0}, n_{+1}\right) \mid \boldsymbol{N}\right] \\ &= 0 + C\left[\frac{1}{n_{+1}} n_{+1}\frac{N_{11}}{N_{+1}}, \frac{1}{n_{+0}} n_{+0}\frac{N_{10}}{N_{+0}} \mid \boldsymbol{N}\right] = 0. \end{aligned} \tag{35}$$

Combining Formulas (33), (34) and (35) with (32) gives:

$$\begin{aligned} V[\hat{\alpha}_c \mid \boldsymbol{N}] &= \frac{N_{+1}^2}{N^2}\frac{1}{n\hat{\alpha}^*}\left[1 + \frac{1-\hat{\alpha}^*}{n\hat{\alpha}^*}\right] \frac{N_{11}N_{01}}{N_{+1}^2} \\ &\quad + \frac{N_{+0}^2}{N^2}\frac{1}{n(1-\hat{\alpha}^*)}\left[1 + \frac{\hat{\alpha}^*}{n(1-\hat{\alpha}^*)}\right] \frac{N_{00}N_{10}}{N_{+0}^2} + O(n^{-3}) \\ &= \frac{1}{n\hat{\alpha}^*}\left[1 + \frac{1-\hat{\alpha}^*}{n\hat{\alpha}^*}\right] \frac{N_{11}N_{01}}{N^2} \\ &\quad + \frac{1}{n(1-\hat{\alpha}^*)}\left[1 + \frac{\hat{\alpha}^*}{n(1-\hat{\alpha}^*)}\right] \frac{N_{00}N_{10}}{N^2} + O(n^{-3}). \end{aligned}$$

Recall from Formula (31) that $\mathrm{V}\left[\hat{\alpha}_c\right] = \mathrm{E}\left[V[\hat{\alpha}_c \mid \boldsymbol{N}]\right] = \mathrm{E}\left[\mathrm{E}\left[V[\hat{\alpha}_c \mid \boldsymbol{N}] \mid N_{+1}\right]\right]$. Hence,

$$\begin{aligned} V[\hat{\alpha}_c] = E\Bigg[&\frac{1}{n\hat{\alpha}^*}\left(1 + \frac{1-\hat{\alpha}^*}{n\hat{\alpha}^*}\right) E\left(\frac{N_{11}N_{01}}{N^2} \mid N_{+1}\right) \\ &+ \frac{1}{n(1-\hat{\alpha}^*)}\left(1 + \frac{\hat{\alpha}^*}{n(1-\hat{\alpha}^*)}\right) E\left(\frac{N_{00}N_{10}}{N^2} \mid N_{+1}\right)\Bigg] + O(n^{-3}). \end{aligned} \tag{36}$$

To evaluate the expectations in this expression, we observe that, conditional on the column total N_{+1}, N_{11} is distributed as $Bin(N_{+1}, c_{11})$, where c_{11} is a calibration probability as defined in Section 2.5. Hence,

$$\begin{aligned} E\left[N_{11} \mid N_{+1}\right] &= N_{+1}c_{11} = \frac{N_{+1}\alpha p_{11}}{(1-\alpha)(1-p_{00}) + \alpha p_{11}} \\ V\left[N_{11} \mid N_{+1}\right] &= N_{+1}c_{11}(1-c_{11}). \end{aligned} \tag{37}$$

Similarly, since $N = N_{+1} + N_{+0}$ is fixed,

$$\begin{aligned} E\left[N_{00} \mid N_{+1}\right] &= N_{+0}c_{00} = \frac{N_{+0}(1-\alpha)p_{00}}{(1-\alpha)p_{00} + \alpha(1-p_{11})} \\ V\left[N_{00} \mid N_{+1}\right] &= N_{+0}c_{00}(1-c_{00}). \end{aligned} \tag{38}$$

Using these results, we obtain:

$$\begin{aligned} E\left[\frac{N_{11}N_{01}}{N^2} \mid N_{+1}\right] &= \frac{1}{N^2}E\left[N_{11}N_{01} \mid N_{+1}\right] \\ &= \frac{1}{N^2}E\left[N_{11}(N_{+1} - N_{11}) \mid N_{+1}\right] \\ &= \frac{1}{N^2}\left[N_{+1}E\left[N_{11} \mid N_{+1}\right] - E\left[N_{11}^2 \mid N_{+1}\right]\right] \\ &= \frac{1}{N^2}\left[N_{+1}E\left[N_{11} \mid N_{+1}\right] - V\left[N_{11} \mid N_{+1}\right] - E\left[N_{11} \mid N_{+1}\right]^2\right] \\ &= \frac{1}{N^2}\left[N_{+1}^2 c_{11} - N_{+1}c_{11}(1-c_{11}) - N_{+1}^2 c_{11}^2\right] \\ &= \frac{N_{+1}^2}{N^2}c_{11}(1-c_{11}) + O\left(\frac{1}{N}\right), \end{aligned} \tag{39}$$

and similarly

$$E\left[\frac{N_{00}N_{10}}{N^2} \mid N_{+1}\right] = \frac{N_{+0}^2}{N^2}c_{00}(1-c_{00}) + O\left(\frac{1}{N}\right). \tag{40}$$

Substituting expressions (39) and (40) into (36) and noting that $N_{+1}^2/N^2 = (\hat{\alpha}^*)^2$ and $N_{+0}^2/N^2 = (1-\hat{\alpha}^*)^2$, we obtain:

$$\begin{aligned} V[\hat{\alpha}_c] = E\Bigg[&\frac{\hat{\alpha}^*}{n}\left(1 + \frac{1-\hat{\alpha}^*}{n\hat{\alpha}^*}\right)c_{11}(1-c_{11}) \\ &+ \frac{1-\hat{\alpha}^*}{n}\left(1 + \frac{\hat{\alpha}^*}{n(1-\hat{\alpha}^*)}\right)c_{00}(1-c_{00})\Bigg] + O\left(\max\left[\frac{1}{n^3}, \frac{1}{Nn}\right]\right) \\ = &\left[\frac{E(\hat{\alpha}^*)}{n} + \frac{1-E(\hat{\alpha}^*)}{n^2}\right]c_{11}(1-c_{11}) \\ &+ \left[\frac{1-E(\hat{\alpha}^*)}{n} + \frac{E(\hat{\alpha}^*)}{n^2}\right]c_{00}(1-c_{00}) + O\left(\max\left[\frac{1}{n^3}, \frac{1}{Nn}\right]\right). \end{aligned}$$

Finally, substituting the expressions for $E(\hat{\alpha}^*)$ from (4) and the expressions for c_{11} and c_{00} from (37) and (38), the desired expression (17) is obtained. This concludes the proof of Theorem 3.

Comparing mean squared errors

To conclude, we present the proof of Theorem 4, which essentially shows that the mean squared error (up to and including terms of order $1/n$) of the calibration estimator is lower than that of the misclassification estimator.

Proof (of Theorem 4). Recall that the bias of $\hat{\alpha}_p$ as an estimator for α is given by

$$B\left[\hat{\alpha}_p\right] = \frac{p_{00} - p_{11}}{n(p_{00} + p_{11} - 1)} + O\left(\frac{1}{n^2}\right).$$

Hence, $(B\left[\hat{\alpha}_p\right])^2 = O(1/n^2)$ is not relevant for $\widetilde{MSE}[\hat{\alpha}_p]$. It follows that $\widetilde{MSE}[\hat{\alpha}_p]$ is equal to the variance of $\hat{\alpha}_p$ up to order $1/n$. From (14) we obtain:

$$\widetilde{MSE}[\hat{\alpha}_p] = \frac{1}{n}\left[\frac{(1-\alpha)p_{00}(1-p_{00}) + \alpha p_{11}(1-p_{11})}{(p_{00} + p_{11} - 1)^2}\right]. \tag{41}$$

Recall that $\hat{\alpha}_c$ is an unbiased estimator for α, i.e., $B[\hat{\alpha}_c] = 0$. Also recall the notation $\beta = (1-\alpha)(1-p_{00}) + \alpha p_{11}$. It follows from (17) that the variance, and hence the MSE, of $\hat{\alpha}_c$ up to terms of order $1/n$ can be written as:

$$\begin{aligned}\widetilde{MSE}[\hat{\alpha}_c] &= \frac{1}{n}\left[\beta\frac{\alpha p_{11}}{\beta}\left(1 - \frac{\alpha p_{11}}{\beta}\right) + (1-\beta)\frac{(1-\alpha)p_{00}}{1-\beta}\left(1 - \frac{(1-\alpha)p_{00}}{1-\beta}\right)\right] \\ &= \frac{\alpha(1-\alpha)}{n}\left[\frac{(1-p_{00})p_{11}}{\beta} + \frac{p_{00}(1-p_{11})}{1-\beta}\right]. \end{aligned} \tag{42}$$

To prove Expression (18), first note that

$$\frac{(1-p_{00})p_{11}}{\beta} + \frac{p_{00}(1-p_{11})}{1-\beta} = \frac{(1-p_{00})p_{11} + \beta(p_{00} - p_{11})}{\beta(1-\beta)}. \tag{43}$$

The numerator of this equation can be rewritten as follows:

$$\begin{aligned}&(1-p_{00})p_{11} + \beta(p_{00} - p_{11}) \\ &= (1-p_{00})p_{11} + (1-\alpha)p_{00}(1-p_{00}) + \alpha p_{00}p_{11} - (1-\alpha)(1-p_{00})p_{11} - \alpha p_{11}^2 \\ &= (1-\alpha)p_{00}(1-p_{00}) + \alpha p_{00}p_{11} + \alpha(1-p_{00})p_{11} - \alpha p_{11}^2 \\ &= (1-\alpha)p_{00}(1-p_{00}) + \alpha p_{11}(1-p_{11}).\end{aligned}$$

Note that the obtained expression is equal to the numerator of Expression (41). Write $T = (1-\alpha)p_{00}(1-p_{00}) + \alpha p_{11}(1-p_{11})$ for that expression. It follows that

$$\begin{aligned}&\widetilde{MSE}[\hat{\alpha}_p] - \widetilde{MSE}[\hat{\alpha}_c] \\ &= \frac{T}{n(p_{00} + p_{11} - 1)^2} - \frac{T\alpha(1-\alpha)}{n\beta(1-\beta)} \\ &= \frac{T}{n(p_{00} + p_{11} - 1)^2\beta(1-\beta)}\Big[\beta(1-\beta) - \alpha(1-\alpha)(p_{00} + p_{11} - 1)^2\Big].\end{aligned}$$

Writing out the second factor in the last expression gives the following:

$$
\begin{aligned}
&\beta(1-\beta) - \alpha(1-\alpha)(p_{00}+p_{11}-1)^2 \\
&= (1-\alpha)^2 p_{00}(1-p_{00}) + \alpha(1-\alpha)\Big((1-p_{00})(1-p_{11}) + p_{00}p_{11}\Big) + \alpha^2 p_{11}(1-p_{11}) \\
&\qquad - \alpha(1-\alpha)(p_{00}+p_{11}-1)^2 \\
&= (1-\alpha)^2 p_{00}(1-p_{00}) + \alpha(1-\alpha)\Big(p_{00}(1-p_{00}) + p_{11}(1-p_{11})\Big) + \alpha^2 p_{11}(1-p_{11}) \\
&= (1-\alpha) p_{00}(1-p_{00}) + \alpha p_{11}(1-p_{11}) \\
&= T.
\end{aligned}
$$

This concludes the proof of Theorem 4.

Deep, dimensional and multimodal emotion recognition using attention mechanisms

Jan Lucas, Esam Ghaleb, and Stylianos Asteriadis[0000−0002−4298−6870]

Department of Data Science & Knowledge Engineering, Maastricht University,
Paul-Henri Spaaklaan 1, 6229 EN Maastricht, Netherlands
jan-lucas@hetnet.nl
{esam.ghaleb, stelios.asteriadis}@maastrichtuniversity.com

Abstract. Emotion recognition is an increasingly important sub-field in artificial intelligence (AI). Advances in this field could drastically change the way people interact with computers and allow for automation of tasks that currently require a lot of manual work. For example, registering the emotion a subject expresses for a potential advert. Previous work has shown that using multiple modalities, although challenging, is very beneficial. Affective cues in audio and video may not occur simultaneously, and the modalities do not always contribute equally to emotion. This work seeks to apply attention mechanisms to aid in the fusion of audio and video, for the purpose of emotion recognition using state-of-the-art techniques from artificial intelligence and, more specifically, deep neural networks. To achieve this, two forms of attention are used. Embedding attention applies attention on the input of a modality-specific model, allowing recurrent networks to consider multiple input time steps. Bimodal attention fusion applies attention to fuse the output of modality-specific networks. Combining both these attention mechanisms yielded CCCs of 0.62 and 0.72 for arousal and valence respectively on the RECOLA dataset used in AVEC 2016. These results are competitive with the state-of-the-art, underlying the potential of attention mechanisms in multimodal fusion for behavioral signals.

Keywords: Emotion Recognition · Multimodal · Neural Networks · Attention Mechanisms

1 Introduction

Emotion recognition as a field in machine learning tries to automate the identification of emotion in a human subject through various means, including computer vision, signal processing and deep learning. This problem is non-trivial as emotion in itself is an abstract concept that is hard to interpret, and thus many different models have been proposed to describe it [9]. Interpretation of emotional expressivity is hindered by the differences in expression between cultures and even persons. Emotion recognition related experiments and data, often come in the form of two types: acted and spontaneous. In the former category, emotions are many times expressed by an actor, while the spontaneous category

mostly involves video clips of spontaneously expressed emotions. Spontaneous emotion recognition is deemed to be harder since it deals with more genuine expression of emotion, which tends to be more subtle than in the acted case.

The state-of-the-art approaches for emotion recognition make use of multiple modalities. This entails that emotion will be predicted by looking at multiple sources. Emotion can be expressed by, for example, both facial and vocal expression, and taking both of these sources into account leads to better performing models [7, 11, 4]. However, using multiple modalities is challenging. These modalities usually differ in multiple aspects, such as their inherent distributions, synchronization, sampling rate, dimensionality, etc.

This work aims to make use of the effectiveness of transfer learning by utilizing pre-trained networks on multiple modalities and fusing these using attention mechanisms. For this purpose, a new method is proposed, that combines previous work in emotion recognition and neural attention to predict emotions in the valence-arousal emotion spectrum.

1.1 Related Work

Work by Ghaleb et al. [4] proposed a framework, which uses metric learning and combines multiple input modalities that showed an increase in performance for discrete emotion recognition in an acted setting [4]. This work relies on some of the preprocessing techniques developed and tested by Ghaleb et al., since it is expected that they will also be beneficial for continuous emotion recognition. Research in the field of emotion recognition is encouraged by the Audio Visual Emotion Challenge (AVEC), which is a competition that is held yearly as part of the ACM Multimedia conference. The competition features a continuous affect recognition sub-part that sets a benchmark for work in the emotion recognition field. The accompanying baseline uses support vector machines (SVM) for each of the eight used modalities, with modality specific feature extraction and processing. SVMs are subsequently fused with a linear regression model [10]. This work will focus on solving the problem presented in this sub-part using feature embeddings and deep neural networks, which have been shown to be effective in similar works [10, 12, 3, 6]. Furthermore, Wu et al. showed that using only feed-forward networks and attention can achieve results that are comparable to recurrent approaches [11]. The method proposed in this work relies on the architectures used by Zhao et al.[12] and Haifeng et al.[3], in which they show that using stacked LSTMs followed by a dense linear layer per modality provides good results. Brady et al. showed state-of-the-art results on the RECOLA dataset for AVEC 2016 using a Kalman Filter approach to fuse models trained for specific modalities [2].

2 Methods

2.1 Attention

Attention mechanisms consitute a family of techniques that can be applied to selectively focus on parts of a sequence. It was initially proposed by Luong et

al. [8] for the purpose of machine translation using an encoder-decoder network. Here the encoder network encodes the sentence in the source language and the decoder uses this encoding to reproduce the sentence in the target language. The order of the words in the source and target languages are often not aligned, and thus the decoder needs to be able to process the encoding out of order. Attention mechanisms allow this network to selectively focus on relevant parts of the encoding to produce a part of the target sequence. The attention mechanism, specifically the general-dot-product (GDP) variant, calculates an attention vector a_t over each encoder hidden state $\overline{h}_s$, which determines how important each part of the input is. This mechanism is formulated as follows:

$$a_t(s) = \frac{\exp(\text{score}(h_t, \overline{h}_s)}{\sum_{s'} \exp(\text{score}(h_t, \overline{h}_{s'})} \tag{1}$$

$$\text{score}(h_t, \overline{h}_s) = h_t^\top W_a \overline{h}_s \tag{2}$$

Here $h_t^\top$ is the current state of the decoder and matrix W_a is a parameter that is to be learned. The attention vector a_t can then be used to compute a weighted average of the source hidden states [8]. Similar to translation, emotions and their corresponding cues in the data may not be aligned. Thus the use of attention mechanisms could improve the performance of models for emotion recognition by allowing them to focus on the information that is relevant for recognizing emotion.

2.2 Proposed approach

The proposed model follows the works of Zhao et al. and Haifeng et al. [12, 3]. These works successfully use LSTMs for multi-modal affect recognition using features that are similar to the ones used in this work. The model used here is a stacked LSTM with dropout between the layers combined with a dense linear layer, where each model outputs both valence and arousal for each time step. This architecture is called the deep long short-term memory network (DLSTM). In total, the network consists of two LSTM layers followed by a dense linear layer. The first LSTM layer iterates over the input sequence and produces a hidden state for each time step in the input. These hidden states are the input for the second LSTM layer, which in turn produces a new sequence of hidden states. These final hidden states are the input to the linear layer that maps them into the two emotion dimensions. This architecture is expanded with attention-based layers on the embedding level (over time) and the fusion level (over the DLSTM networks). Figure 1 shows the overall network structure.

Embedding attention Attention over the embeddings closely follows the general dot-product (GDP) approach described by Luong et al. for language translation using encoder-decoder architectures [8]. The GDP attention mechanism computes scores for each source hidden state $\overline{h}_s$ in a sequence. A softmax function is applied to these to obtain a distribution which is subsequently used to

construct a weighted combination of the sequence. GPD attention computes the score in the following way: $h_t^\top W_a \overline{h}_s$. Where h_t is the current state of the decoder and $\overline{h}_s$ is a hidden state of the encoder. W_a is a weight matrix that maps the encoder hidden state and the target hidden state to a score and has to be optimized. To adapt this method for affect recognition, the target hidden state h_t is the hidden state of the DLSTM, C_v in figure 1, and the source states are replaced with the embedding vectors in a certain window of size n, represented by $V_{t-\frac{n}{2}} \cdots V_{t+\frac{n}{2}}$ in figure 1. The attention mechanism thus computes the score of an embedding vector depending on the current state of the DLSTM and the contents of the embeddings.

This attention mechanism is applied only to the video modality. As will be explained in section 3, the extracted audio features already contain temporal information and should therefore benefit less from attention.

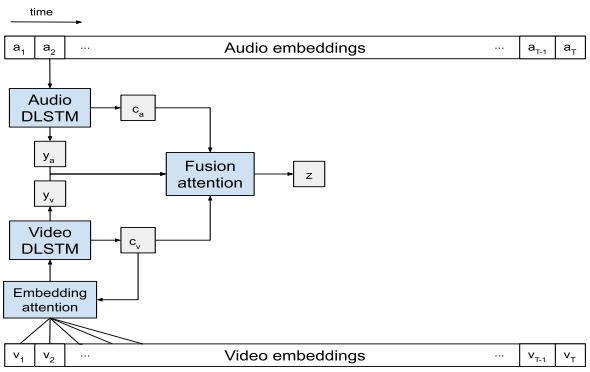


Fig. 1. Overview of the proposed model. a_t and v_t respectively, are the audio and video embeddings at time t. The variables y_m and c_m are the output and hidden state of the DLSTM responsible for modality m, and z represents the fused output.

Bimodal attention fusion The attention concept is also applied to decision fusion. The information that the DLSTMs receive may not allow for a very good prediction of emotion at every time step. For example, at some time t, audio may be more appropriate for emotion recognition, whereas the visual signal may not be carrying significant information. In that case, the output of the audio LSTM should have a higher weight in decision making. GDP attention mechanism outputs a linear combination of the inputs, resulting in equal weight being given to the valence and arousal output of a DLSTM. This is not necessarily the best solution, as at some time step t the audio DLSTM might be very "confident" about its output for arousal and unsure for valence, but the video DLSTM may be confident about valence. The GDP attention mechanism cannot assign different weights for each output dimension separately.

We experimented with an attention approach to decision fusion: Bimodal Attention Fusion. This method uses attention to combine the outputs of modality-specific DLSTM networks. Attention scores are computed by considering all DLSTM hidden states at once. Here $C \in \mathbb{R}^{cn}$, with c being the size of the hidden states and n the number of DLSTMs, is a vector containing the concatenation of the hidden states of all the underlying DLSTMs at time t. $Y_m \in \mathbb{R}^n$ is a vector containing the outputs of the DLSTMs for output modality m. This vector contains either valence or arousal outputs of all the DLSTM networks. The scores and outputs are computed separately for both arousal and valence, because different output dimensions may require different distribution of attention. The calculation of the scores can be reformulated as follows:

$$S_m = C^\top W_m \tag{3}$$

$$a_{m_i} = \frac{\exp(s_i)}{\sum_{j=1}^{n} \exp(s_j)} \tag{4}$$

$$Z = \{z_1, \cdots, z_o\}, \text{where } z_m = \sum_{i=1}^{n} a_{m_i} y_{m_i} \tag{5}$$

The attention vector A, whose elements are described in equation 4, is the matrix product of the DLSTM hidden states in C and the learned weights in W_m followed by an application of the softmax function. The weights $W_m \in \mathbb{R}^{cn \times n}$ map the hidden states to scores per DLSTM and are optimized separately per output modality m. This allows for different mappings from hidden states to scores for both valence and arousal. The attention weights in A_m are used to take a weighted combination of the DLSTM outputs.

Baseline fusion methods The effectiveness of the attention mechanism is assessed by comparing it to networks that fuse without attention. The first network realizes fusion as a static combination of the network outputs by a dense linear layer and is named output linear baseline (OLB). When $Y \in \mathbb{R}^{o \times n}$ is a matrix containing the LSTM outputs, with o being the number of output dimensions, and $W \in \mathbb{R}^{n \times 1}$ is a weight matrix, the fused output can be formulated as follows:

$$Z = YW \tag{6}$$

Matrix W is optimized during training and linearly fuses the outputs of the DLSTMs. The second network fuses by using the concatenation of the hidden states of the DLSTMs. This method, named hidden linear baseline (HLB), can be formulated as:

$$Z = C^\top W \tag{7}$$

Just as in equation 3, $C \in \mathbb{R}^{cn}$ is a matrix containing the concatenation of the hidden states of the DLSTMs, but here $W_m \in \mathbb{R}^{cn \times o}$ is a parameter that maps the hidden states directly to the output emotions.

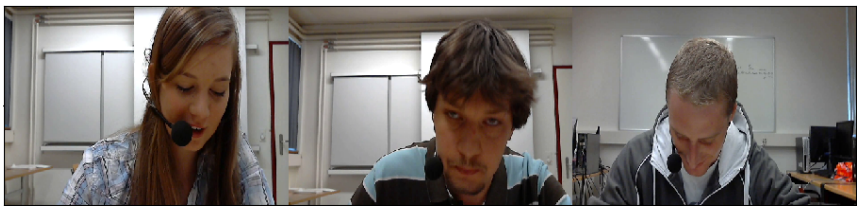


Fig. 2. Example frames from the RECOLA dataset

3 Results and discussion

This section details the experiments performed to determine the performance of the architecture described in section 2.2. A subset of the RECOLA dataset from the University of Fribourg was used in the 2015 and 2016 Audio/Visual Emotion Challenge (AVEC), and is used in this work to train the proposed architecture. The dataset consists of 18 five minute clips in predetermined train and validation partitions [10]. Examples of frames from this dataset are shown in figure 2. This subset of RECOLA contains several feature types, but only the raw video and audio data is used. Emotion is annotated continuously in the valence-arousal space for each video frame. Face extraction is performed on each frame of video and the raw data is transformed using embedding networks. Activations from the last convolutional layer of the VGGFace network is used to extract frame-level video features and VGGish is used to extract audio features from a 960ms window [1, 5].

The models mentioned below are trained by optimizing the mean squared error using the Adam optimizer with a learning rate of 0.01. The DLSTM architecture, described in section 2.2, is optimized using truncated backpropagation through time, following Zhao et al. and Haifeng et al. [12, 3]. The training and test partitions provided with the RECOLA data were used in order to make a fair comparison with the results from the state of the art from AVEC. Model performance is assessed with the Concordance Correlation Coefficient (CCC) measure, which is a common measure of performance in emotion recognition. The CCC is computed for each sequence in test partition of the dataset and averaged over the sequences to show the performance.

3.1 Attention

As explained in section 2.2, attention mechanisms are applied on two points in the proposed model. Embedding attention, which applies attention to the embeddings used as input to the video DLSTM, and fusion attention, which uses attention to fuse the outputs of the DLSTMs. These two methods are evaluated separately in the following experiments.

Embedding attention To assess the effectiveness of applying attention to the video embeddings, the performance of the DLSTM is compared with and

without attention. For embedding attention, a window size of 13 frames is used, allowing the mechanism to consider a segment of half a second. The baseline in this comparison is a DLSTM without any attention, and thus processes the embeddings sequentially, instead of being able to focus on a window.

Initial performance of the model with embedding attention was very poor and stagnated directly after start of training. This was in contrast with the behavior of the model without embedding attention and suggested that training was hindered by the addition of the attention mechanism. To counteract this, the embedding attention layer is bypassed for the first five training epochs. After this startup period, the embedding layer is included again, which significantly improved performance. A possible explanation for this phenomenon is the cyclic dependency between the embedding attention and the hidden layer of the DLSTM. The embedding attention layer uses the hidden state to compute the input to the DLSTM, which in turn affects the hidden state. A meaningless hidden layer could result in poorly attended input, which then maintains the form of the hidden state. To account for random initialization, training and testing is repeated 10 times. For arousal, this resulted in very similar results regardless of the use of attention, with CCCs of 0.15 ($\pm$0.05) and 0.14 ($\pm$0.05) for no attention and embedding attention respectively. A slight improvement was observed for valence with CCC results of 0.36 ($\pm$0.07) without attention and 0.39 ($\pm$0.06) with embedding attention. Even though promising, this difference is not statistically significant, with $p > 0.05$. A bidirectional variant of the DLSTM without attention, since the embedding attention allows the network to use frames ahead of the current time step, and a bigger attention window were evaluated, but these resulted in similar CCC scores.

Table 1. CCC results for the pre-trained uni-modal DLSTMs (left) and their fusion using bimodal attention fusion and baselines (right). Both Bimodal attention and Hidden linear baseline (HLB) successfully fuse the unimodal networks for valence prediction.

Uni-modal	Valence	Arousal
audio	0.42	0.60
video	0.24	0.10

Fusion	Valence	Arousal
Bimodal attention	0.48 ($\pm$0.04)	0.60 ($\pm$0.1)
OLB	0.32 ($\pm$0.03)	0.40 ($\pm$0.08)
HLB	0.48 ($\pm$0.01)	0.64 ($\pm$0.02)

Fusion attention Section 2.2 describes the attention mechanism that can be used to combine the outputs of the uni-modal DLSTM networks. To make a fair comparison with the detailed baselines, two DLSTM networks are pre-trained separately on audio and video, and subsequently frozen before training the fusion mechanisms. This procedure restricts the performance of the model as a whole, but allows for a clear comparison of the fusion methods. The training of the fusion mechanisms is repeated 10 times while using the same pre-trained DLSTMs, to account for random initialization.

The results can be seen in figure 3 and table 1. The CCC values for the pre-trained network are also detailed in table 1. Comparing the methods shows that the proposed attention fusion mechanism significantly improves performance when compared to fusion by linearly combining the DLSTM outputs (OLB). However, its performance is matched by the baseline method that regresses the hidden states of the DLSTMs directly (HLB). For valence, the HLB baseline and bimodal attention mechanism both showed better performance than the unimodal networks they fused. This suggests that the performance of the audio network is slightly increased by fusion with the video modality, but the difference is not large enough for any concrete conclusions. In short, the proposed method seems fuse the uni-modal networks successfully, however its performance does not improve beyond the HLB baseline, which does not use attention.

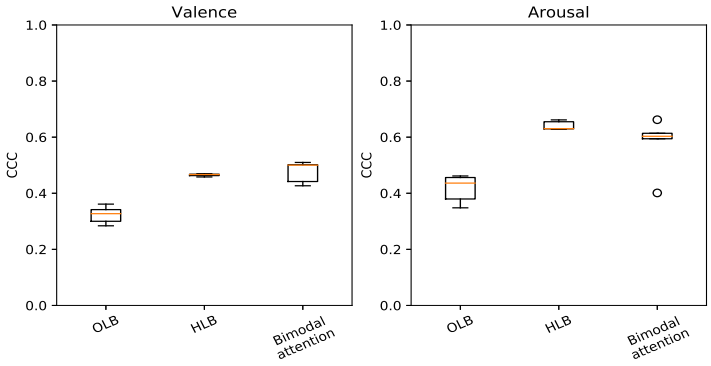


Fig. 3. CCC results for fusion of pre-trained uni-modal DLSTM models using bimodal attention fusion and baseline methods.

3.2 Comparison with the state of the art

The previous sections have each evaluated parts of the proposed architecture. In this section, a comparison with recent works from the literature is performed. For this, the network is trained in an end-to-end fashion using the bimodal attention fusion method and embedding attention on the video modalty. Hyperparameters are optimized empirically, resulting in hidden sizes of 32 and 128 for the audio and video DLSTMs respectively. Work by Haifeng et al. [3] shows that early fusion of features combined with decision level fusion provides improved results for emotion recognition. For this purpose, audio and video features are concatenated per time step to form an early fusion modality. The model described in section 2.2 is extended with a third DLSTM, with a hidden size of 128, that is used on this new modality. The outputs of this DLSTM are fused with the outputs from the audio and video DLSTMs to produce the final model output. The results are compared with a baseline provided by AVEC [10] and the best results on this dataset, achieved by Brady et al. [2]. This comparison is displayed in table 2. The CCC scores obtained by the proposed model for arousal are higher

than the baseline, but are below the ones by Brady et al. However, it should be highlighted here that these two works make use of a wider set of modalities (such as electrodiograms and electrodermal activity, beyond just video and audio), whereas, in the proposed method, only audio and video are considered. For valence, the performance is just below the baseline. The results obtained with the proposed model are comparable to the results of these methods, even though fewer modalities were used and embedding techniques from other domains were reused.

Table 2. Performance of the model proposed in section 2.2 compared to the AVEC baseline and state of the art for this dataset.

	Valence	Arousal
Baseline	0.683	0.639
State of the art (Brady et al.)	0.702	0.82
Proposed	0.62	0.72

4 Conclusions and future work

This work explored combining the information in audio and video data by using attention to fuse the output of networks that were trained on only one modality each. Furthermore, attention was used to spot important video embeddings in temporal windows using the hidden state of an LSTM network.

Fusion of the output modalities using attention shows a significant improvement when compared to a model that does not take the states of the input networks into account. However, it shows similar performance to the baseline that directly regresses the hidden states, suggesting that more improvements should be possible. Usage of embedding attention showed promising results, but this difference is not significant, with a p-value greater than 0.05. Applying attention on the embedding level produced new challenges, that were overcome by using a special training procedure. Future work could investigate the causes for this and explore other, more flexible, variants of this mechanism.

Combining embedding attention and fusion attention yields a model that shows promising performance. Results exhibited improved performance compared to the AVEC baseline for arousal and close performance for valence. The proximity to the baseline and state-of-the art results shows the potential of the proposed method, since the baseline and state-of-the-art methods use more modalities and fine-tuned pre-processing methods. This is in contrast to the proposed method, which uses fewer modalities and reuses feature embeddings from other domains. Other modalities can be easily included in the proposed method and it is expected that this will improve results.

In conclusion, the use of attention mechanisms for emotion recognition shows promising results and can successfully combine information from multiple modalities. Future research could expand on this architecture by experimenting with different forms of attention, extra modalities and different feature embeddings.

References

1. Barsoum, E., Zhang, C., Ferrer, C.C., Zhang, Z.: Training deep networks for facial expression recognition with crowd-sourced label distribution. In: Proceedings of the 18th ACM International Conference on Multimodal Interaction. p. 279–283 (2016). https://doi.org/10.1145/2993148.2993165
2. Brady, K., Gwon, Y., Khorrami, P., Godoy, E., Campbell, W., Dagli, C., Huang, T.S.: Multi-modal audio, video and physiological sensor learning for continuous emotion prediction. In: Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge. p. 97–104. AVEC '16, Association for Computing Machinery (2016). https://doi.org/10.1145/2988257.2988264
3. Chen, H., Deng, Y., Cheng, S., Wang, Y., Jiang, D., Sahli, H.: Efficient spatial temporal convolutional features for audiovisual continuous affect recognition. In: Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop. p. 19–26. AVEC '19, Association for Computing Machinery (2019). https://doi.org/10.1145/3347320.3357690
4. Ghaleb, E., Popa, M., Asteriadis, S.: Metric learning based multimodal audio-visual emotion recognition. IEEE MultiMedia pp. 1–1 (2019). https://doi.org/10.1109/MMUL.2019.2960219
5. Hershey, S., Chaudhuri, S., Ellis, D.P.W., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., Slaney, M., Weiss, R.J., Wilson, K.: CNN architectures for large-scale audio classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 131–135
6. Huang, J., Tao, J., Liu, B., Lian, Z., Niu, M.: Efficient modeling of long temporal contexts for continuous emotion recognition. In: 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII). pp. 185–191 (9 2019). https://doi.org/10.1109/ACII.2019.8925452
7. Kossaifi, J., Walecki, R., Panagakis, Y., Shen, J., Schmitt, M., Ringeval, F., Han, J., Pandit, V., Toisoul, A., Schuller, B.W., Star, K., Hajiyev, E., Pantic, M.: SEWA DB: A rich database for audio-visual emotion and sentiment research in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2019)
8. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1412–1421. Association for Computational Linguistics (Sep 2015). https://doi.org/10.18653/v1/D15-1166
9. Poria, S., Cambria, E., Bajpai, R., Hussain, A.: A review of affective computing: From unimodal analysis to multimodal fusion. Information Fusion **37** (02 2017). https://doi.org/10.1016/j.inffus.2017.02.003
10. Valstar, M., Gratch, J., Schuller, B., Ringeval, F., Lalanne, D., Torres Torres, M., Scherer, S., Stratou, G., Cowie, R., Pantic, M.: Avec 2016: Depression, mood, and emotion recognition workshop and challenge. In: Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge. p. 3–10. AVEC '16, Association for Computing Machinery (2016). https://doi.org/10.1145/2988257.2988258
11. Wu, Z., Zhang, X., Zhi-Xuan, T., Zaki, J., Ong, D.C.: Attending to emotional narratives. In: 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII). pp. 648–654. IEEE Computer Society (sep 2019). https://doi.org/10.1109/ACII.2019.8925497
12. Zhao, J., Li, R., Chen, S., Jin, Q.: Multi-modal multi-cultural dimensional continues emotion recognition in dyadic interactions. In: Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop. p. 65–72. AVEC'18, Association for Computing Machinery (2018). https://doi.org/10.1145/3266302.3266313

A Spiking Neuron Implementation of Genetic Algorithms for Optimization

Siegfried Ludwig[1], Joeri Hartjes[1], Bram Pol[1], Gabriela Rivas[1], and Johan Kwisthout[2][0000−0003−4383−7786]

[1] School for Artificial Intelligence, Radboud University
Montessorilaan 3 6525 HR Nijmegen, Netherlands
siegfried.m.ludwig@protonmail.ch
[2] Donders Center for Cognition, Radboud University
Montessorilaan 3 6525 HR Nijmegen, Netherlands
j.kwisthout@donders.ru.nl

Abstract. We designed freely scalable ensembles of spiking neurons to carry out the operations required to run a genetic algorithm, thereby opening up possibilities for making use of efficient neuromorphic hardware. Two types of implementation are explored that offer a complexity trade-off between computational space and time, with both designs having linear energy complexity. The designs were implemented in a simulator to successfully solve the one-max optimization problem, serving as a proof of concept for running genetic algorithms as spiking neural networks.

Keywords: neuromorphic computing · genetic algorithm · spiking neural networks

1 Introduction

Neuromorphic computing ranges back to the term being coined in 1990 [1], in which the first implementation consisted of very large scale integration (VLSI) with analog components mimicking the biological neural systems. Much research has been done since this time, and in the last few years the energy efficiency of such architectures have become an increasingly dominant research subject. Spiking neural networks (SNN) are known as a type of neuromorphic implementation which have exceptional energy saving properties, compared to other systems [2]. SNNs augment artificial neural networks with the spiking dynamics found in biological neurons [3]. Based on leaky integrate-and-fire (LIF) neurons [4], SNNs transmit information by means of timing and energy spikes, released when the potential difference inside a neuron reaches a certain threshold. This is because such hardware is modeled after the brain in that its activation is event-driven and asynchronous. On top of that, SNN's property of local information storage effectively avoids the von Neumann bottleneck arising from an idling processor while retrieving data from memory [5]. Researching the possibility of implementing various existing algorithms in such SNNs leads the way to a future in which

real life applications of such algorithms currently implemented on von Neumann architectures could be replaced.

Implementing algorithms as SNNs to run them on neuromorphic hardware has been done for sorting [6], constraint satisfaction [7], shortest path and neighborhood subgraph extraction problems [8]. The striking similarity between a genetic sequence and a neural spike train inspires the implementation of a genetic algorithm (GA) as a SNN, which could make use of recent hardware developments in neuromorphic computing.

The use of evolution-inspired algorithms has been proven a viable solution for tackling problems of optimization, bringing in advantages for optimisation over traditional methods. For instance, GA systems [9, 10] may provide the opportunity for difficult problem solving such as multi-objective optimisation [11] and have found applications in various practical settings (see [12] for a review). As in natural evolution, GAs work by modifying the characteristics of individuals in a population across several iterations. This is done by means of reproduction (*crossover*) and random gene mutation. With each run, individuals with an arrangement of genes with a higher fitness value are allowed to preferentially reproduce and carry over their genetic information into the next *generation*. In this study, each individual solution (*chromosome*) is represented as a binary bit sequence, in which each bit represents the value of a gene.

Our main aim was to investigate the feasibility of implementing a GA using spiking neurons with the potential for future implementation on neuromorphic hardware, such as Intel's Loihi [13] or IBM's TrueNorth [14]. Our design consists of binary genetic sequences, which are represented as neuronal spikes and are processed by LIF neurons with context-dependent parameters. The chosen optimization problem is the *one-max problem* due to its simplicity and wide use in the literature on genetic algorithms; the objective of which is to produce a fully active genetic sequence, in this case a fully active spike train. The neural network was implemented and tested using a spiking neuron simulator[3].

We considered two candidate possibilities for encoding the binary genetic sequence in neural ensembles. Firstly, the genetic sequence can be represented sequentially as a spike train, with a spike indicating a 1 and no spike indicating a 0. An ensemble in this design processes one bit at a time. The second way of representing a binary genetic sequence is parallel, using a separate neuron for each position of the genetic sequence. These two encodings are expected to offer a complexity trade-off between computational space and time.

In the following, the high-level architecture of the SNN is presented, followed by details on the sequential and parallel implementations. We then conduct a complexity analysis of both implementations with regard to space, time and energy, in order to assess the tractability of our design.

[3] https://gitlab.socsci.ru.nl/j.kwisthout/neuromorphic-genetic-algorithm

2 High-Level Architecture

The genetic algorithm consists of initializing and evaluating a starting population and then repeatedly performing selection, crossover, mutation, and evaluation on the population until termination. It is implemented as a single recurrent SNN, consisting of specialized neural ensembles for each operation (Figure 1). The topology of the network gives a fitness hierarchy, with the fittest chromosomes being at the top and conversely the least fit chromosomes being at the bottom. The network architecture is static during run-time and no learning of the weights is required.

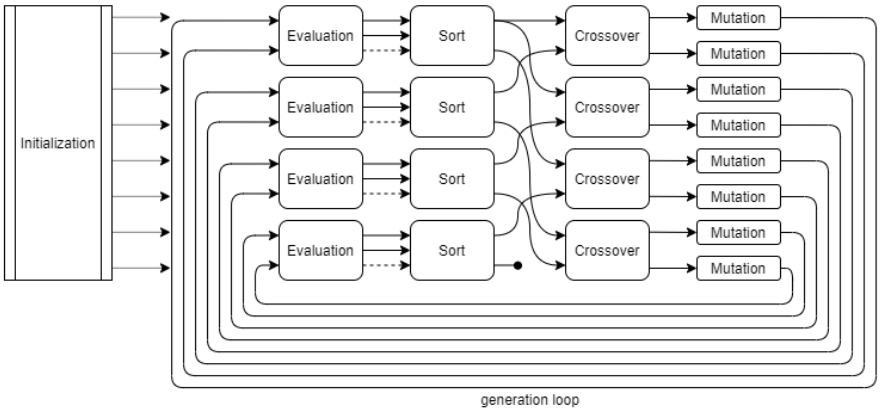


Fig. 1: High-level architecture of the genetic algorithm network, depicted with 8 chromosome lanes as solid arrows (each arrow can represent multiple neural connections in the parallel design). After *mutation*, all chromosomes are connected back to *evaluation* to close the generation loop, resulting in a single large neural network. The dashed arrows from *evaluation* to *sort* represent the evaluation result. To increase the number of chromosomes, the pattern of the second pair of the four lane pairs is repeated.

After initialization, the chromosomes enter the evaluation ensembles in pairs, where they are evaluated against each other and then potentially swapped to bring the chromosome with higher fitness to the top. This setup corresponds to a single pairwise bubble sort step and over time ranks chromosomes by their fitness, which is necessary for selection. The use of only a limited number of bubble sort steps in each generation will lead to incomplete sorting, but is more efficient and leads to some variety in the ranking of the chromosomes while still avoiding the removal of very promising individuals from the bottom. This design is related to the ranking selection mechanism [15].

Selection is implemented in the connections from the sorting ensembles to the crossover ensembles, by eliminating the bottom chromosome and connecting

the top chromosome twice. This results in better solutions propagating more successfully over time. In addition to potentially moving up one lane in the sorting ensemble itself, the winner of the pairwise evaluations moves up by another lane after sorting to ensure upwards mobility, as otherwise the same pairs would be compared each iteration. Conversely, the inferior chromosome moves down by a number of lanes after sorting.

Crossover is then performed on each pair of chromosomes. This reproduction is implemented with a stochastic crossover method, which splits two sequences at a random point and swaps all subsequent genes between the individuals.

After crossover, each chromosome is processed individually in a mutation ensemble. Mutation is carried out by assigning a probability for flipping the activity of each bit in a given sequence. In our designs, we use a probability of $p = \frac{1}{n}$, where n is the length of the chromosome, but other mutation rates are possible. We do not apply mutation on the top two chromosomes of each generation in order to allow for stable one-max solutions. Again, this choice is more up to the design of the genetic algorithm than the implementation as a spiking neural network.

To close the generation loop, the outputs of the mutation ensembles connect back into the evaluation ensembles, forming a recurrent neural network.

Scaling the network up for a larger population size or longer chromosomes is straight-forward beyond a small minimum size, by repeating whole ensembles and repeated elements within certain ensembles.

3 Neural Ensembles for Genetic Algorithms

3.1 Sequential Design

In our sequential design of the GA, the chromosome is processed one bit at a time, which more closely resembles genetic processing in nature. Sequential processing allows a small neural ensemble to process arbitrary lengths of chromosomes over time without itself growing in size. The implementation relies on a lead bit, which precedes every chromosome and is always active. This allows the signaling of the arrival of a new chromosome, ensuring correct processing. In the following, neurons will be ascribed different types based on their function in the ensemble. However, they are all based on the LIF neuron model.

Evaluation Ensemble The sequential one-max evaluation ensemble (Figure 2) makes use of 8 neurons and 11 internal connections. It takes as input two chromosomes and gives as output two chromosomes as well as a spike on a separate neuron serving as an indicator in case the bottom chromosome has a higher fitness than the top chromosome. The membrane potential of the accumulator neuron (ACC) is increased with each active bit in the bottom chromosome, and decreased with each active bit in the top chromosome. Note that the ACC neuron is like all other neuron types used here just a LIF neuron with specific parameters. The activation neuron (A), activated by the lead bit, then makes the ACC

neuron fire or not based on the final membrane potential of the ACC neuron. Only in the case of a membrane potential higher than zero will the indicator neuron fire, and will the chromosomes' ranking switch. A reset neuron (R) is responsible for spiking but suppressing the ACC as to prevent interference of previous chromosome comparisons with current iterations. Clearing the potential of the ACC neuron could alternatively be done using membrane leakage over time, but that would result in a less predictable design.

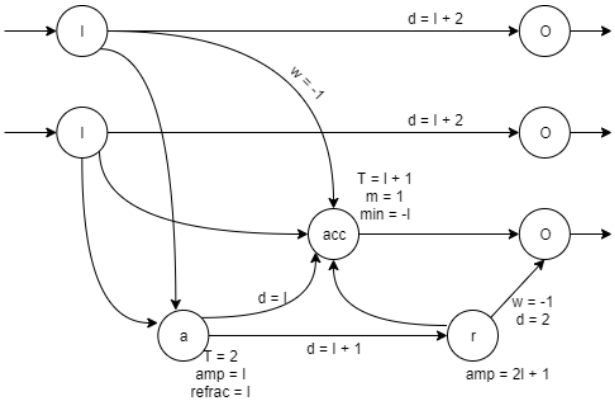


Fig. 2: Sequential one-max evaluation ensemble. (I) Input, (acc) Accumulator neuron, (a) Activation neuron, (r) Reset neuron, (o) Output.

Bubble Sort Ensemble The bubble sort ensemble (Figure 3) consists of 10 neurons and 15 internal connections. It takes two chromosomes plus a fitness indication as input and gives two chromosomes as output. It uses gate (G) neurons to open or close the identity and swap lanes connecting input and output and thereby controlling whether the incoming chromosomes are swapped or propagated as identity. This is achieved by giving the swap gate neurons a threshold of two, which means they can only fire if an input comes from the gate control (GC) neuron. The GC neuron is activated by the gate control activation (GCA) neuron, which takes the fitness indicator input coming from the evaluation ensemble. The GC neuron uses a recurrent connection to keep the swap gates open and the identity gates closed until the chromosomes passed through entirely, at which point it is deactivated by a delayed spike coming from the GCA neuron.

Crossover Ensemble The crossover ensemble (Figure 4) works similarly to the bubble sort ensemble, except that identity and swap gates are not open or closed for the whole chromosome, but switch activation at a random point. It uses 13 neurons and 27 internal connections. The ensemble could be simplified to only use one gate control (GC) neuron as in the bubble sort ensemble, but

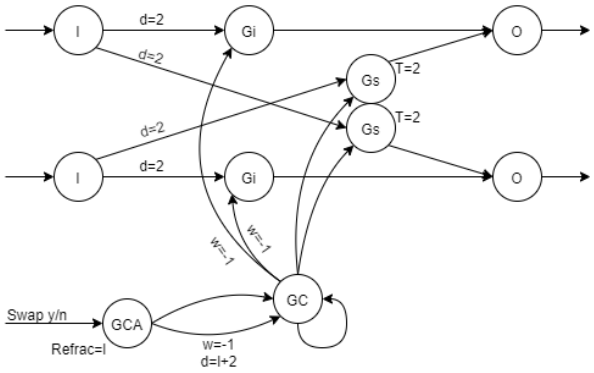


Fig. 3: Bubble sort ensemble in the sequential design. (I) input, (Gi) identity gate, (Gs) swap gate, (GC) gate control, (GCA) gate control activation, (O) output.

has been implemented with two in this project. The random crossover point is implemented via a stochastic (S) neuron and a stochasticity control (SC) neuron. The S neuron gets constant input from the SC neuron, while also generating a random membrane potential each time step. If this combined potential crosses the S neuron's threshold the identity GC neuron is deactivated and the swap GC neuron is activated. If S spikes, The S neuron also deactivates the SC neuron, since only one crossover point is desired.

Mutation Ensemble Finally, the mutation ensemble (Figure 5) stochastically turns a 0 into a 1 and conversely a 1 into a 0, independently for each bit excluding the lead bit. It uses 6 neurons and 12 internal connections. The first stochastic neuron (S1) gets a positive input from each spike in the input and adds a random membrane potential, which can cross the threshold and lead to a spike. A spike from S1 suppresses the ensemble output, thereby turning a 1 into a 0. The other stochastic neuron (S2) always gets input from the control (C) neuron and adds a random membrane potential, but it is suppressed by every spike in the input. If no spike comes from the input, it has a chance of firing and turning the ensemble output from a 0 to a 1. The control neuron is activated and finally deactivated by the control activation (CA) neuron.

Full Network Behavior Each chromosome is passed through the ensembles in its lane, as described in the high-level architecture (see Figure 1). In the sequential design, a chromosome can still be processed in one ensemble while already entering into the next (e.g. crossover to mutation), since here each bit can be handled independently. An exception to this is the evaluation ensemble, which needs to accumulate the full chromosome to make an evaluation. It therefore breaks the time-constant flow through the other ensembles and leads to a time

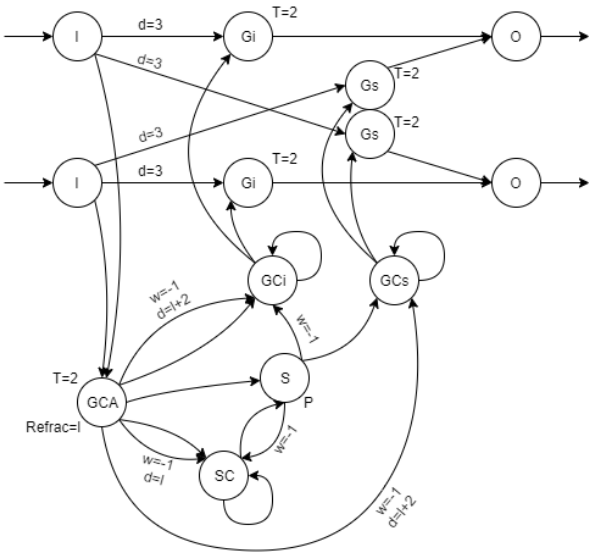


Fig. 4: Crossover ensemble in the sequential design. (I) input, (Gi) identity gate, (Gs) swap gate, (GCi) identity gate control, (GCs) swap gate control, (GCA) gate control activation, (S) stochastic, (SC) stochasticity control, (O) output.

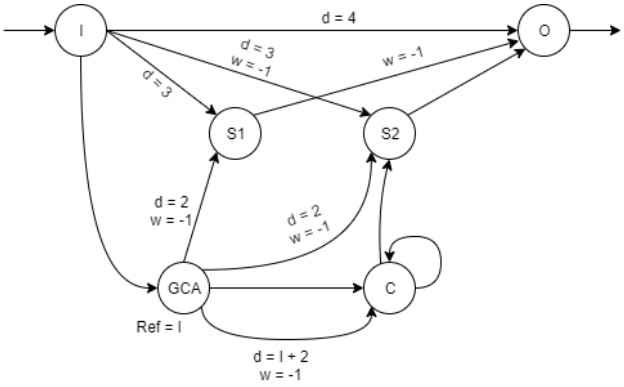


Fig. 5: Mutation ensemble in the sequential design. (I) input, (S1) stochastically flips 1 to 0, (S2) stochastically flips 0 to 1, (C) constant, (GCA) gate control activation, (O) output.

dependency on the chromosome length. On the upside this prevents chromosomes from being longer than the execution cycle, which could otherwise lead to the beginning of the next generation interfering with the end of the last for long chromosomes.

3.2 Parallel Design

In the parallel implementation, every gene of the chromosome gets processed at the same time. Instead of using a single spike train to represent the chromosome, multiple neurons are used that each represent one gene of the chromosome. A set of neurons can then represent the binary code of the chromosome by either spiking or not. Its advantage is that the entire binary code of the chromosomes can be conveyed in a single time step, but requires more neurons as chromosomes get longer. A generation of the entire algorithm in parallel design takes exactly eleven simulation time steps. Again, all neuron types presented here are simple LIF neurons with specific parameters. The different ensembles used in the algorithm will be explained below.

Evaluation & Bubble Sort Ensemble In the parallel design the evaluation step is combined with the bubble sort step. The goal of the evaluation is to have the chromosome with the highest fitness be transferred to the first n output neurons where n is the length of the chromosome. One lead bit is present for each chromosome pair which enables functionality in the other ensembles of the GA, however for this segment it is of no use and therefore linked directly to its corresponding output neuron. Also for this reason the decision was made to omit the lead bit altogether in Figure 6. By taking advantage of all information contained in the chromosome being available at once, the evaluation and sorting ensembles could be combined. This enables the comparison of the fitness through the use of one Accumulator neuron (ACC) to which all input genes are connected (excluding the lead bit). The sign of the connection weights leading to the ACC results in it becoming active only if the lower chromosome has a greater fitness than the top chromosome by at least one gene. Subsequently, the activation of the ACC will determine whether either the identity gates or the swap gates are activated. These are responsible for transferring the activity from the input to the output neuron of the same, or the 'adversarial chromosome', respectively.

Each of the input neurons are connected to both a dedicated identity gate and a dedicated swap gate, with these being connected to the identity neuron or the neuron on the other chromosome in the same position. The connection from the input to the gates is delayed by one time step, however, to allow for synchronous arrival of the spike and the spike coming from the ACC. The connections between the ACC and the gates are weighted such that by default the identity gates have a threshold low enough that a spike from the input neurons will be enough to spike the gate as well while the threshold of the swap gates is too high. As soon as the ACC is activated however this spike is no longer enough for the identity gates, while the extra activation coming from the ACC to the swap gates lowers

their threshold enough to let the spike pass from the input neuron to the correct output neuron on the side of the 'adversarial chromosome'.

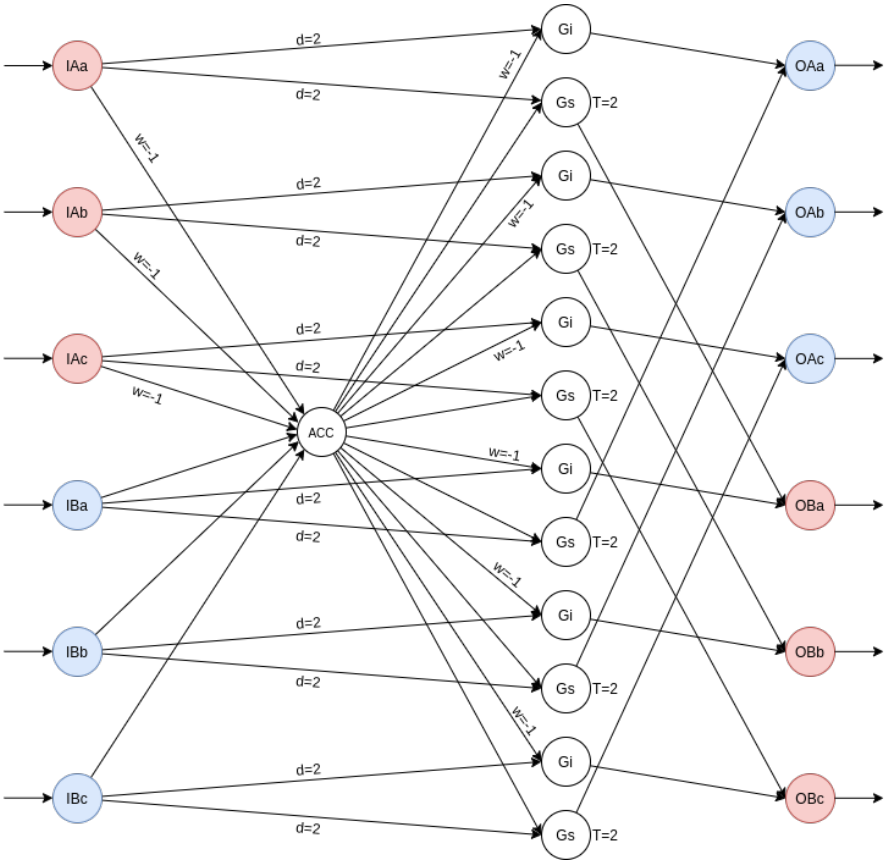


Fig. 6: Ensemble responsible for the evaluation and sorting in the parallel design, applied to a pair of chromosomes consisting of three genes each. Using an accumulator neuron (ACC), the ensemble determines which of the chromosomes has higher fitness and places the winner in the top lanes.

Crossover Ensemble The parallel crossover ensemble can be seen in Figure 7. The first gene of every chromosome always ends up in the same output chromosome. The last gene is always crossed over and ends up in the opposite chromosome. To decide where the genes in between go, a 'random point maker' has been designed (see Figure 8), which is activated by the lead bit. The input to the second layer of the random point maker spikes with a probability of $p = \frac{1}{n-2}$,

where n is the chromosome length. If activated, the node in this second layer transfers this spike to all nodes in the third layer on the same level or below, ensuring that once a gate opens the gates below also open.

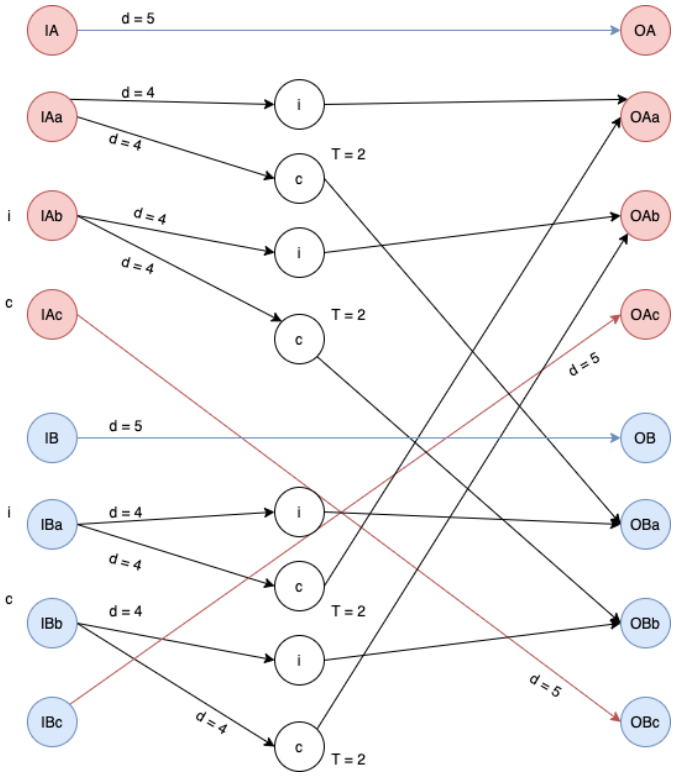


Fig. 7: The parallel crossover ensemble. The first gene of a chromosome is always sent to the same position and the last gene is always crossed over. For the genes in the middle, the random point maker determines whether they are crossed over or not.

The third layer of the random point maker (the gates) connect to the identity and crossover nodes in the crossover ensemble. When a gate neuron of the random point maker gets a spike, it closes the identity gate and opens the crossover gate of both chromosomes at that level. This way, initially the crossover ensemble will transfer genes to the same output chromosome, but at a random point will switch to crossing genes over to the other output chromosome. The crossover ensemble, together with the random point maker, takes five time steps to run for any chromosome length n. The number of neurons in the ensemble is $10n - 11$, meaning linear growth. The number of connections does not show linear growth,

because the connections between the second and third layer of the random point maker grow with $\frac{n^2+n}{2}$, which is quadratic growth. Because of this, the number of connections in the whole ensemble grows quadratically.

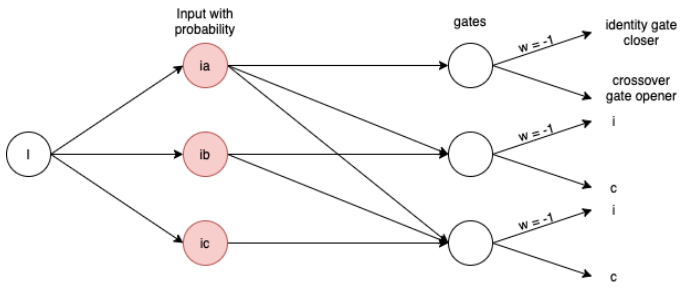


Fig. 8: The random point maker that connects to the gates of the crossover ensemble. This ensemble determines the point of the chromosomes where the identity ends and the swapping of genes with each other starts. It makes sure that there is an equal probability for every point in the chromosome to be the start of crossing over the remaining genes.

Mutation Ensemble The final ensemble in the parallel design is responsible for the stochastic mutation of the genes in the chromosomes, meaning turning a 1 into a 0 or vice versa (Figure 9). The way it is implemented is through assigning a probability P to each of the genes, and therefore neurons, to switch their activity. Except for the lead bit (Ia), every input-neuron (Ib, Ic) is connected to two neurons, and both of their thresholds are influenced by the switching-probability through $T = 2 - P$. A noise factor is present in both intermediate neurons, its function being to add randomness as to whether a neuron will mutate or not. In the diagram the first of the two intermediate neurons is responsible for potentially turning off the activation in case that the input neuron has spiked, and the other is responsible for the opposite. Each of the input neurons is connected directly to its corresponding output neuron, however this connection is delayed such that its spike is delivered synchronously to the potential spike of one of the two intermediate neurons. The role of the lead bit is essential to the mutation ensemble, as its guaranteed activity allows for the potential activation of the two intermediate neurons, which otherwise have no chance of reaching their threshold. Combining the stochastic nature of the intermediate neurons, together with the configuration of the intermediate neurons then has the desired effect of a random mutation of the gene together with the appropriate switching of its value.

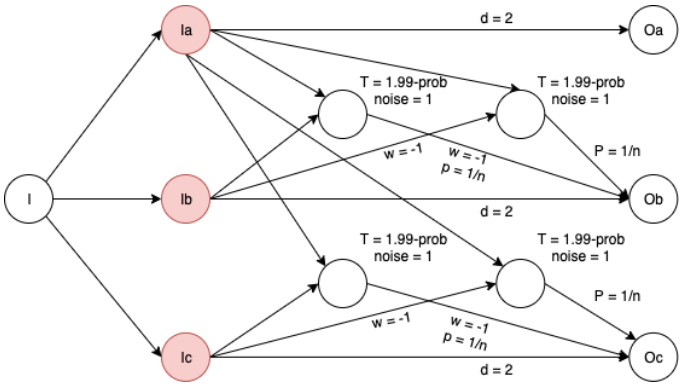


Fig. 9: The parallel mutation ensemble. This ensemble makes sure that every gene's bit has a chance to be swapped.

4 Analysis

A raster plot of the output neurons of the sequential bubble sort ensemble is given in Figure 10. It shows the improving solution quality over time, with the top chromosome reaching one-max, and also shows some resemblance of the fitness hierarchy, with better solutions being closer to the top (subject to imperfect sorting). Figure 11 confirms that the top chromosome in the hierarchy has a higher than average fitness, which specifically shows that even the single pairwise bubble sort step at each generation is enough to at least approximate a fitness ranking.

To assess the tractability of our two designs, a complexity analysis is performed. Computational complexity for neuromorphic computing is considered in terms of space, time, and energy, measured as the number of spikes. For this analysis, all three complexities have been considered with regard to the number of chromosomes and the chromosome length. Comparing the complexity of the sequential and parallel design shows a space-time trade-off between the two (Table 1), with the sequential design requiring less space but more time. Both designs have linear space complexity in the number of chromosomes, both in terms of the number of neurons and the number of connections. The sequential design has much lower space requirements however.

Regarding the chromosome length, the sequential design has constant space complexity, while the parallel design is linear in the number of neurons and quadratic in the number of connections. This is the least favorable of all measured behaviors. It is specifically caused by the current implementation of randomly determining a crossover point. Both designs have constant time complexity in the number of chromosomes, with time measured in simulation steps per generation.

While the parallel design also has constant time complexity in the chromosome length, the sequential design has linear time complexity. The sequential design inherently needs to have at least linear time complexity in the chromo-

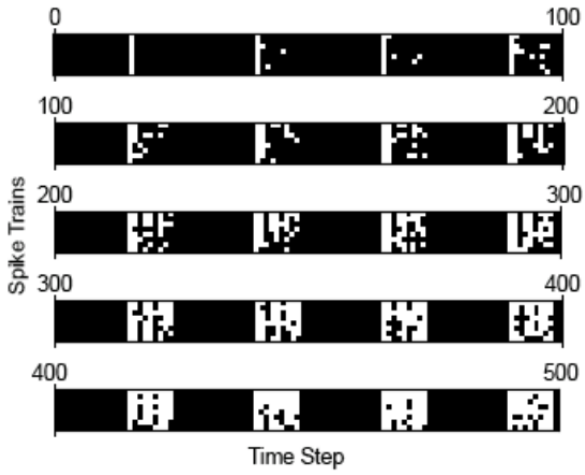


Fig. 10: Raster plot of bubble sort output neurons over time in the sequential design (8 chromosomes of length 8, for 500 steps). Each row of pixels depicts a neural spike train over 500 simulation steps. The solution quality is improving over time, with the top chromosome reaching one-max.

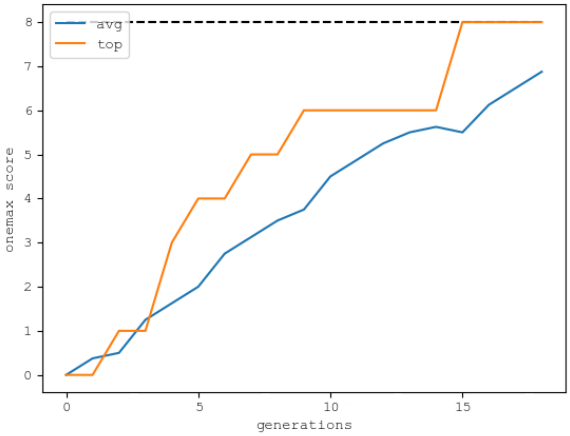


Fig. 11: Average and best solution quality over generations (8 chromosomes of length 8, for 500 steps). The fitness hierarchy results in the top chromosome having better fitness than the average. While this plot comes from the sequential design, the parallel design behaves similarly.

Table 1: Complexity analysis of space, time, and energy (number of spikes) for both the sequential and the parallel design. There is a trade-off between space and time comparing the two designs, with the sequential design requiring less space but more time.

		n_chromosomes		len_chromosomes	
		sequential	parallel	sequential	parallel
space	neurons	$O(n)$	$O(n)$	$O(1)$	$O(n)$
	connections	$O(n)$	$O(n)$	$O(1)$	$O(n^2)$
time		$O(1)$	$O(1)$	$O(n)$	$O(1)$
energy		$O(n)$	$O(n)$	$O(n)$	$O(n)$

some length, as the full chromosome needs to be accessed before an evaluation can be made. This is an advantage for the parallel design, as the full chromosome is available at once. Both designs have linear energy complexity in the number of chromosomes and in the chromosome length, when measuring energy as the average number of spikes required to process one generation.

5 Discussion

A fully functioning genetic algorithm has been successfully implemented as a spiking neural network with two different designs, representing chromosomes sequentially as a spike train over time or as parallel spikes at a single time step. Both implementations are freely scalable beyond a small minimum number of chromosomes, with arbitrary chromosome lengths. The complexity analysis of space, time and energy shows the tractability of this approach with the exception of the quadratically growing number of connections required for the parallel design when increasing chromosome length. The sequential design is at most linear in any of the analyzed complexities.

The design has not yet been implemented on neuromorphic hardware. Since fairly standard leaky integrate-and-fire neurons were used, however, and no learning is required, translating the design to an implementation in neuromorphic hardware should be relatively straight-forward.

For future work, the design of the crossover ensembles could be adapted to support gene lengths of more than one bit (a chromosome consists of a number of genes, which itself could consist of a number of bases/bits). Practically this just means that the random crossover point should only be allowed at transition points between genes, so at fixed intervals. This would allow for more complex behavior of the genetic algorithm.

More work needs to be done on the evaluation strategy, which under the current design requires a unique neural ensemble purpose-built for the optimization task at hand and thereby presents a hurdle for practical application. One possibility for a more general approach would be to train a spiking neural network to

perform approximate evaluations for the given task, instead of hand-engineering the neural ensemble for exact solutions as is performed in this paper.

References

[1] Carver Mead. "Neuromorphic electronic systems". In: *Proceedings of the IEEE* 78.10 (1990), pp. 1629–1636.

[2] Catherine D Schuman et al. "A survey of neuromorphic computing and neural networks in hardware". In: *arXiv preprint arXiv:1705.06963* (2017).

[3] Samanwoy Ghosh-Dastidar and Hojjat Adeli. "Spiking neural networks". In: *International journal of neural systems* 19.04 (2009), pp. 295–308.

[4] Anthony N Burkitt. "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input". In: *Biological cybernetics* 95.1 (2006), pp. 1–19.

[5] Aaron R Young et al. "A review of spiking neuromorphic hardware communication systems". In: *IEEE Access* 7 (2019), pp. 135606–135620.

[6] Samya Bagchi, Srikrishna S Bhat, and Atul Kumar. "O(1) time sorting algorithms using spiking neurons". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2016, pp. 1037–1043.

[7] Chris Yakopcic et al. "Solving constraint satisfaction problems using the loihi spiking neuromorphic processor". In: *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2020, pp. 1079–1084.

[8] Catherine D Schuman et al. "Shortest path and neighborhood subgraph extraction on a spiking memristive neuromorphic implementation". In: *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*. 2019, pp. 1–6.

[9] John H Holland. "Genetic algorithms". In: *Scientific american* 267.1 (1992), pp. 66–73.

[10] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press, 1992.

[11] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons, 2001.

[12] Manoj Kumar et al. "Genetic algorithm: Review and application". In: *Available at SSRN 3529843* (2010).

[13] Chit-Kwan Lin et al. "Programming spiking neural networks on Intel's Loihi". In: *Computer* 51.3 (2018), pp. 52–61.

[14] Paul A Merolla et al. "A million spiking-neuron integrated circuit with a scalable communication network and interface". In: *Science* 345.6197 (2014), pp. 668–673.

[15] L Darrell Whitley et al. "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best." In: *Icga*. Vol. 89. Fairfax, VA. 1989, pp. 116–123.

Reputation-driven Decision-making in Networks of Stochastic Agents

David Maoujoud[1] and Gavin Rens[2]

[1] Katholieke Universiteit Leuven, Belgium
david.maoujoud@student.kuleuven.be, david.maoujoud@hotmail.com
[2] Katholieke Universiteit Leuven, Belgium
gavin.rens@kuleuven.be

Abstract. This paper studies multi-agent systems that involve networks of self-interested agents. We propose a Markov Decision Process-derived framework, called RepNet-MDP, tailored to domains in which agent reputation is a key driver of the interactions between agents. The fundamentals are based on the principles of RepNet-POMDP, a framework developed by *Rens et al.* [11] in 2018, but addresses its mathematical inconsistencies and alleviates its intractability by only considering fully observable environments. We furthermore use an online learning algorithm for finding approximate solutions to RepNet-MDPs. In a series of experiments, RepNet agents are shown to be able to adapt their own behavior to the past behavior and reliability of the remaining agents of the network. Finally, our work identifies a limitation of the framework in its current formulation that prevents its agents from learning in circumstances in which they are not a primary actor.

Keywords: Uncertainty · Planning · Reputation · MDP · POMDP.

1 Introduction

Decision-making and learning in multi-agent settings is a multi-faceted area of research [4, 2, 3, 14, 7, 6, 1]. Frameworks used for fully cooperative networks of agents differ vastly from those used for networks of self-interested agents. A primary concern when dealing with self-centered agents is that it makes multi-agent learning inherently more complex than single-agent learning [6, 1]. In fact, each agent needs to take into account the behavior of the entire network of agents when learning its own behavior. Additionally, agent behavior tends to be ever-changing. This *non-stationarity* of agent behavior leads to the loss of policy *convergence* properties that can often be found in single-agent formalisms [1].

In 2018, *Rens et al.* [11] proposed a mathematical framework, called *RepNet-POMDP*, designed to handle partially observable environments in which an agent's reputation among other agents dictates its behavior. The framework was subject to several mathematical inconsistencies, had no working implementation, and had a highly intractable planning algorithm.

Nonetheless, the framework does present some ideas we believe are worth pursuing. Hence, in this paper, we provide an updated version of the framework, called RepNet-MDP. We address the mathematical inconsistencies of the original framework and alleviate its intractability by only considering fully observable environments. We furthermore make use of an online learning algorithm for finding approximate solutions to RepNet-MDPs. The viability of the framework is tested in a series of experiments designed to highlight its strengths and shortcomings.

Section 2 summarizes the relevant background required. Section 3 gives an overview of the work related to our framework. Section 4 provides an intuitive introduction to RepNet-MDPs. Section 5 covers the formal definition of the framework. Section 6 covers planning for RepNet-MDPs. The experimental setup and results are given in Section 7.

2 Background - Markov Decision Processes

A Markov Decision Process (MDP) describes a process for modeling decision-making in stochastic environments [13]. An agent is assumed to move about in an environment, described by a set of states $\mathcal{S}$, by applying actions in $\mathcal{A}$ to the environment. The transition rules of the environment are dictated by the transition model $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, that is, $\mathcal{T}(s, a, s')$ returns the probability of the agent transitioning to state s' upon performing action a in state s. Each action applied to the environment results in a reward for the agent, dictated by the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, that is, $\mathcal{R}(s, a)$ returns the reward received by the agent when performing action a in state s.

The objective of an MDP agent is to maximize its long-term cumulative reward, called *utility*. The utility U of a *finite* state-action sequence, sometimes called *episode*, $E = \langle s_0, a_0, s_1, a_1, ..., s_T, a_T \rangle$ is defined as [10]:

$$U(E) = \sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t),$$

where $\gamma \in [0, 1]$ is called the *discount factor*. An agent advances in the environment by following a *policy* $\pi : \mathcal{S} \times \mathbb{N} \to \mathcal{A}$ that maps each environment state and remaining time-steps to the action the agent should take.

The *expected utility*, or *value*, of being in any state s_t at time-step t, while following policy π, with d time-steps remaining, is defined as:

$$V^\pi(s_t, d) = \mathbb{E}[U(E_t) \mid s_t, \pi] = \mathbb{E}\big[\sum_{k=t}^{t+d} \gamma^{k-t} \mathcal{R}(s_k, a_k) \,\big|\, s_t, \pi\big],$$

where E_t is the sub-sequence of E starting at time-step t. An optimal policy $\pi^\star$ is a policy such that

$$\forall s \in \mathcal{S}, \forall d \in \mathbb{N}, \forall \pi : V^\star(s, d) \geq V^\pi(s, d),$$

where $V^\star : \mathcal{S} \times \mathbb{N} \to \mathbb{R}$ is the value function associated with optimal policy $\pi^\star$. This policy satisfies the *optimality equations*, also known as the *Bellman equations* ($\forall s \in \mathcal{S}$):

$$\begin{cases} V^\star(s,d) := \max\limits_{a \in \mathcal{A}} \Big\{ \mathcal{R}(s,a) + \gamma \sum\limits_{s' \in \mathcal{S}} \mathcal{T}(s,a,s') V^\star(s', d-1) \Big\} & d > 1 \\ V^\star(s,1) := \max\limits_{a \in \mathcal{A}} \Big\{ \mathcal{R}(s,a) \Big\} \end{cases}$$

Partially Observable Markov Decision Processes are a common extension of classic MDPs that deal with the problem of partial observability of the environment [13]. To address the agent's inability to observe the exact state of the environment, the observation function $\mathcal{O} : \mathcal{A} \times \mathcal{S} \times \Omega \to [0,1]$, where Ω is the set of observations, is introduced. $\mathcal{O}(a, s', o)$ returns the probability of the agent making observation o after performing action a and the environment transitioning to state s'.

Instead of working with the actual states of the environment, the POMDP agents make use of the notion of belief state $b \in \Delta(\mathcal{S})$ [3], which is a probability distribution over the possible states of the environment. Belief states are updated using the *state estimation function* SE defined as follows:

$$b' := SE(b,a,o) := \Big\{ (s', p) \;\Big|\; s' \in \mathcal{S} \wedge p = \frac{\mathcal{O}(a,s',o) \sum_s \mathcal{T}(s,a,s') b(s)}{P(o|b,a)} \Big\},$$

where $P(o|b,a) = \sum_{s' \in \mathcal{S}} \mathcal{O}(a,s',o) \sum_{s \in \mathcal{S}} \mathcal{T}(s,a,s') b(s)$ is a normalizing constant. We refer to [13] for an extensive overview of POMDPs.

3 Related MDP-based frameworks

Early multi-agent frameworks, such as Multi-agent Markov Decision Processes (MMDPs) [4] and Decentralized Partially Observable MDPs (Dec-POMDPs) [2, 3], operate under the assumption that the agents are selfless and have a common goal. Consequently, planning can be centralized, that is, each agent's policy can be computed by central unit, before being distributed amid the agents for execution [14]. Dec-POMDPs furthermore differ from MMDPs in that states are no longer fully observable, meaning that each agent is in possession of its own set of *local observations*.

In 2005, *Gmytrasiewicz et al.* formalized an extension of POMDPs to multi-agent settings, called Interactive-POMDP (I-POMDP) [7]. I-POMDPs are designed for reasoning in networks of selfish agents. I-POMDP agents update their beliefs not only over physical states of the environment but also over models of the other agents in the network. The difficulty of solving I-POMDPs lies in the recursive nature of the models. Consider agent g's belief update in a network inhabited by another agent, say h. A model of agent h may consist of the belief function of said agent h over physical states and models of all other agents.

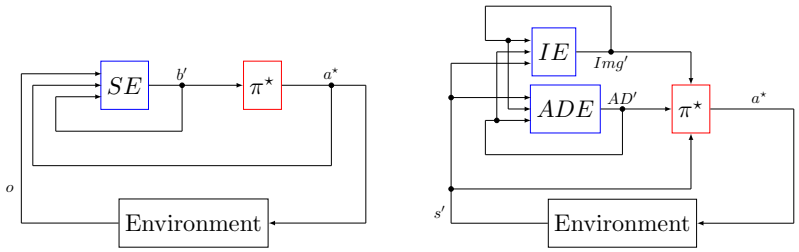
[3] $\Delta(\mathcal{E})$ is the set of probability distributions over the elements of set $\mathcal{E}$.

These models may, in turn, consist of belief functions of their own. This nesting of beliefs could theoretically be infinite, but is overcome by bounding the nesting depth by a finite number n, and solving the problem as a set of POMDPs.

The RepNet-MDP framework [11] simplifies the notion of model by focusing in on key concepts such as behavioral habits and reputation of other agents. While this reduces the insights RepNet agents can have into other agents' behavior, it makes the framework arguably more intuitive. The key, novel notion in the RepNet framework is that of subjective transitions, which have a dependence on the reputation of the agent performing the action.

4 Developing an intuition for RepNet-MDPs

To develop an intuition for the RepNet-MDP framework, parallels between the concepts found in classic POMDPs and RepNet-MDPs can be drawn. In a POMDP, a single agent, placed in a partially observable environment, applies an action $a^\star$ it deems optimal as per its current policy $\pi^\star$, and is sent back an observation o. The state estimation function SE can be thought of as a way of extracting information from said observation o, and storing it in a belief state b'. More specifically, o contains information about the actual state of the environment. The POMDP loop is depicted in Fig. 1a.



(a) The POMDP loop (b) The RepNet-MDP loop

Fig. 1: POMDP and RepNet-MDP loops.

Let us now consider a fully observable environment made up of 3 selfish agents, of which the behavior of the first is dictated by the RepNet-MDP framework. The willingness of the RepNet agent to engage with agents 2 or 3 is to be conditioned by their reputation and behavioral habits. The first agent once again applies action $a^\star$, as per its policy $\pi^\star$. The environment returns its new state s'. In an effort to make well-informed decisions, the RepNet agent should extract the other agents' behavior from s'.

Two functions, analogous to the state estimation function SE in POMDPs, are used to this end: The action distribution estimation function ADE extracts information regarding other agents' behavioral habits. The image estimation function IE informs the RepNet agent on the image all the agents have of each other. The RepNet-MDP loop is shown in Fig. 1b.

Closely tied to the concept of image is the notion of reputation. Specifically, the reputation of any agent in the framework can be seen as a summary of the

information encapsulated by the image. Unlike POMDPs, RepNet-MDPs feature two types of actions, and by extention two types of transition models:

- *Objective* actions, which, when performed, have a real effect on the environment. These actions can be seen as equivalent to actions as they exist in MDPs. The associated transition model is called the objective transition model OT and describes the rules of the environment as they apply to the RepNet agent.
- *Subjective* actions, which, unlike objective actions, are never actually applied to the environment. Instead, they are associated with another transition model called the subjective transition model ST: This transition model describes a RepNet agent's subjective perception of the rules of the environment. This perception is a function of said agent's reputation, and can be used by the agent to aid in its decision-making.

5 Formal definition of RepNet-MDPs

In this section, we will formalize the RepNet-MDP framework[4] introduced in Section 4. A RepNet-MDP $\mathcal{M}$ is defined as a pair of tuples $\mathcal{M} := \langle \Sigma, \Gamma \rangle$, where Σ is called the System tuple and incorporates aspects of the network that apply to all agents, and Γ is called the Agents tuple and contains each agent's subjective understanding of the environment it operates in.

Specifically, a System in a RepNet-MDP Σ is formally defined as a tuple

$$\Sigma := \langle \mathcal{G}, \mathcal{S}, \mathcal{A}, \mathcal{I}, \mathcal{U}, OT \rangle,$$

where:

- $\mathcal{G}$ is the set of agents that can interact with the environment.
- $\mathcal{S}$ is the set of possible states of the environment.
- $\mathcal{A}$ is the set of possible actions, both *objective* and *subjective*. Formally, $\mathcal{A} := \mathcal{A}^o \cup \mathcal{A}^s,\ \mathcal{A}^o \cap \mathcal{A}^s := \varnothing$. The concept of *subjective actions* will be discussed in Section 5.2.
- $\mathcal{I} : \mathcal{G} \times \mathcal{G} \times \mathcal{S} \times \mathcal{A} \to [-1, 1]$ is called the *impact function*. $\mathcal{I}(g, h, s, a)$ returns the impact on agent g that is due to agent h performing action a in state s. This function can be thought of as analogous to a Markov Decision Process's immediate reward function $\mathcal{R}$.
- $\mathcal{U} : [-1, 1] \times [-1, 1] \to [-1, 1]$ is called the *image update function*. Given a current value v, of the image an agent has of another agent, to be updated, and a new expected total impact i, of which the definition will be given shortly, $\mathcal{U}(v, i)$ returns an updated value of the image v'. Many instantiations of this function are possible, two of which are presented in [11]. We will use the following instantiation:

$$\mathcal{U}(v, i) := \begin{cases} v + (1 - v)i & \text{if } i \geq 0 \\ v + (1 + v)i & \text{if } i < 0 \end{cases}$$

[4] The implementation of the RepNet-MDP framework can be found at https://github.com/davidmaoujoud/RepNet-MDP

- $OT : \mathcal{G} \times \mathcal{S} \times \mathcal{A}^o \times \mathcal{S} \to [0,1]$ is called the *objective transition model*. $OT(h,s,a,s')$ returns the probability of the environment transitioning from state s to state s' when objective action a is taken by agent h.

In addition to the global information stored in Σ, each RepNet agent's subjective knowledge is stored in the Agents tuple Γ, formally defined as [5]

$$\Gamma := \big\langle \{ST_g\}, \{AD_g\}, \{Img_g\} \big\rangle,$$

where:

- $ST_g : \mathcal{G} \times \mathcal{S} \times \mathcal{A}^s \times \mathcal{S} \times [-1,1] \to [0,1]$ is called the *subjective transition model* of agent g. $ST_g(h,s,a,r_h,s')$ returns the probability, *as perceived by agent* g, of the environment transitioning from state s to state s' if agent h were to perform subjective action a, and has a reputation r_h according to agent g.
- $AD_g : \mathcal{G} \times \mathcal{S} \to \Delta(\mathcal{A})$ is called the *action distribution* according to agent g. $AD_g(h,s)$ returns a probability distribution over actions in $\mathcal{A}$ for agent h in state s, according to agent g.
- $Img_g : \mathcal{G} \times \mathcal{G} \to [-1,1]$ is called the *image function* according to agent g. $Img_g(h,i)$ returns the image agent i has of agent h according to agent g. Said differently, it returns what g thinks i thinks of h.

As introduced in Section 4, every agent bases its decision-making on the image it believes all agents to have of each other, as well as each agent's behavioral habits. Let g be an agent, whose image at time t is Img_g, and action distribution is AD_g. At time $t+1$, these constructs are updated via the image estimation function IE and action distribution estimation function ADE respectively, to produce Img'_g and AD'_g.

5.1 Image and Reputation

This subsection builds towards the formal definition of the image estimation function IE. To this end, we introduce the notion of *expected total impact*. Consider two agents h and i. In any given state, agent h can perform one of several actions which may or may not have an impact on agent i. Likewise, agent i can be expected to have an impact on agent h when performing an action. The *expected total impact* should be thought of as a way of assigning a numerical value to the *bidirectional* impact these two agents can be expected to have on each other. Additionally, one direction of the impact may be perceived as more important than the other and thus be weighed differently. According to an observing agent, say g, the total impact h is expected to have on, as well as perceive from, i when the environment is in state s, is defined as

$$\begin{aligned} ETI_g(h,i,s,AD_g) := \sum_{a \in \mathcal{A}} \big[& \delta AD_g(i,s)(a)\mathcal{I}(h,i,s,a) \\ & + (1-\delta) AD_g(h,s)(a)\mathcal{I}(i,h,s,a) \big], \end{aligned}$$

[5] $\{X_g\}$ is used as the shorthand notation for $\{X_g \mid g \in \mathcal{G}\}$

where $\delta \in [0, 1]$ weighs the importance of impact due to agent h and impact perceived by h.

Agent g's image of other agents, as well as the image it believes all agents to have of each other changes as it observes the agents' behavior. Let Img_g be the current image function of agent g. Concretely, we wish to *update* the image any agent i has of any other agent h, according to the observing agent g $(= Img_g(h, i))$ on the basis of the impact h is expected to have on i $(= ETI_g(h, i, s, AD_g))$. The updated image function Img'_g is computed as follows:

$$\begin{aligned} Img'_g &:= IE(g, Img_g, \alpha, s, AD_g) \\ &:= \Big\{(h, i, t) \;\Big|\; h, i \in \mathcal{G} \wedge t = \mathcal{U}(Img_g(h, i), ETI_g(h, i, s, AD_g))\Big\}, \end{aligned}$$

where s is the current state of the environment, IE is called the *image estimation function*, and $\mathcal{U}$ is the *image update function*.

Finally, the notion of reputation as it is understood in this framework can be thought of as a way of summarizing the information encapsulated by the image.

Say agent g wishes to estimate the reputation of agent h in a network made up of several other agents. It can, to this end, use the image each agent i has of agent h $(= Img_g(h, i))$ as a guiding principle. A first idea might be to take agent h's reputation to be equal to its average image in the network. If, however, some agent i has a poor image of agent h $(Img_g(h, i) < 0)$, but agent g has a poor image of agent i $(Img_g(i, g) < 0)$, it may be unreasonable for agent g to assume that agent i's opinion of agent h is indicative of agent h's reputation being poor. These concerns are addressed by weighing the image each agent i has of h by the image g has of i. As such, if both images are negative, the resulting reputation of h will not be affected negatively $(Img_g(h, i) \times Img_g(i, g) > 0)$. Formally, the reputation of an agent h, according to agent g, is defined as

$$REP_g(h, Img_g) := \frac{1}{|\mathcal{G}'|} \sum_{i \in \mathcal{G}'} Img_g(h, i) \times Img_g(i, g),$$

where $Img_g(i, i) = 1$ $\forall i \in \mathcal{G}$, and $\mathcal{G}' = \mathcal{G}$ if $h \neq g$ and $\mathcal{G}' = \mathcal{G} \setminus \{g\}$ if $h = g$. Recall that in the RepNet framework, reputation influences subjective transition probabilities, which, in turn, influence a RepNet agent's planning.

5.2 Subjective actions and the subjective transition model

In this section, we describe the use of subjective actions and subjective transition models in the RepNet framework. As introduced in Section 4, we make a distinction between the purpose of an *objective* transition model, which describes the actual rules of the environment as they apply to the RepNet agent, and that of a *subjective* transition model, which describes that agent's *subjective* perception of the rules of the environment, this perception being influenced by the reputation of the RepNet agent. To illustrate this further, we will make use of a simple trading example between two agents A and B. Agent A wishes to

trade with agent B, who can either accept or refuse the trade offer. The environment is made up of the set of states $\mathcal{S} = \{s_0, s_1, s_a, s_r\}$. s_0 is the initial state, prior to any trade, s_1 is the state in which agent B is made aware of agent A's trade offer, s_a is the accept state, and s_r is the refuse state. The set of *objective* actions at the disposal of both agents is given by

$$\mathcal{A}^o = \{\texttt{trade\_with\_A}, \texttt{trade\_with\_B}, \texttt{accept}, \texttt{refuse}, \texttt{wait}\}.$$

The transition model of the environment assumed to be deterministic, is given in Fig. 2. *In the eyes of* agent A, agent B's response to a trade offer, characterized

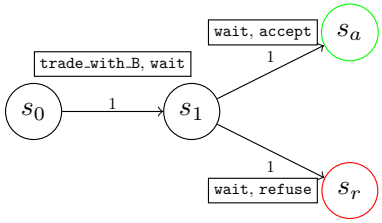


Fig. 2: Transition model of the environment (trading example). Each transition has two *objective* actions, the first action represents agent A's *objective* action, the second action represents agent B's *objective* action.

by transitions $s_1 \to s_a$ and $s_1 \to s_r$, depends on A's reputation. The action taken by agent A during these transitions is `wait`. To make use of the notion of subjective actions, the set of *subjective* actions $\mathcal{A}^s$ will contain the counterpart[6] of `wait` in its *subjective* form, that is, $\mathcal{A}^s = \{\texttt{wait\_s}\}$.

The way agent A makes use of actions in $\mathcal{A}^o$ and $\mathcal{A}^s$ can now be detailed. When *planning* to maximize its expected impact, agent A will make use of the *objective* transition model whenever the action currently investigated has no subjective counterpart in $\mathcal{A}^s$. For instance, the transition probability when investigating action `trade_with_B` is given by $OT(A, s_0, \texttt{trade\_with\_B}, s_1)$. When an action in $\mathcal{A}^o$ has a counterpart in $\mathcal{A}^s$, agent A will make use of the subjective transition model. For instance, the transition probability when investigating action `wait`/`wait_s` is given by $ST_A(A, s_1, \texttt{wait\_s}, s_a, r_A)$. As such, the reputation of agent A is accounted for when agent A *plans* to maximize its expected impact.

5.3 Action distribution

The next step in the formalization of RepNet-MDPs consists in redefining the updating scheme of the action distribution AD_g of each agent g. Say an environment hosting two agents g and h is currently in state s. Agent g has an *a priori* notion of the probability of agent h picking an action a in state s, $P_g(a|h, s, r_h)$. Following agent h performing action a in this state, the environment transitions

[6] We define *counterpart* as a partial mapping $\mathcal{C} : \mathcal{A}^o \to \mathcal{A}^s$. If $\mathcal{C}(a)$ is not defined, then a has no counterpart in $\mathcal{A}^s$.

from state s to state s'. The *a posteriori* probability of agent h performing that same action a in state s in the future is now computed using Bayes' rule:

$$P_g(a|h,s,r_h,s') = \frac{P_g(s'|h,s,r_h,a)P_g(a|h,s)}{\sum_{a'} P_g(s'|h,s,r_h,a')P_g(a'|h,s)}.$$

One can add a *smoothing* technique called *Laplace smoothing* to the present result in an effort to avoid undesirable side effects related to the use of deterministic transition models. The definition for the *action distribution estimation*, obtained after replacing the probabilities by the terms defined previously and applying the smoothing technique, is then given by:

$$\begin{aligned} AD'_g &:= ADE(g, s', AD_g, Img_g) \\ &:= \left\{ (h,s,a,p) \;\middle|\; h \in \mathcal{G} \wedge s \in \mathcal{S} \wedge a \in \mathcal{A} \wedge r_h = REP_g(h, Img_g) \right. \\ &\qquad \left. \wedge\, p = \frac{T_g(h,s,a,s',r_h)AD_g(h,s)(a) + \eta}{\sum_{a'}(T_g(h,s,a',s',r_h)AD_g(h,s)(a') + \eta)} \right\}, \end{aligned}$$

where ADE is called the *action distribution estimation function*, s' is the state the environment transitions to, and η is the *Laplace smoothing parameter*. We refer to the accompanying extensive version of this paper [9] for more details.

Note that to simplify the notation, we combined the *objective* and *subjective* transition models into a single model T_g, called the *global transition model* and formally defined as

$$T_g(h,s,a_h,s',r_h) := \begin{cases} ST_g(h,s,a_h,s',r_h) & \text{if } a_h \in \mathcal{A}^s \\ OT(h,s,a_h,s') & \text{if } a_h \in \mathcal{A}^o \end{cases} \tag{1}$$

6 Planning in the RepNet framework

We now describe optimal behavior in the context of RepNet-MDPs, for finite horizon look-ahead. To simplify the notation, we can define a construct called *epistemic state*. The epistemic state θ_g of agent g is formally defined as a tuple $\theta_g := \langle s, AD_g, Img_g \rangle$, where s is the current state of the environment, AD_g is the current action distribution of agent g, and Img_g is the current image function of agent g. $\theta_g \in \Theta_g$, and Θ_g is called the *epistemic state space*. This set contains every possible combination of physical states of the environment, action distributions, and image functions of agent g.

An agent should perform actions according to the *perceived immediate impact* they have on the agent itself. The perceived immediate impact on agent g resulting from performing action a in state s is defined as

$$PI_g(s, AD_g, a) := \frac{1}{|\mathcal{G}|}\left[\mathcal{I}(g,g,s,a) + \sum_{h \in \mathcal{G}\setminus\{g\}} \sum_{a' \in \mathcal{A}} \mathcal{I}(g,h,s,a')AD_g(h,s)(a')\right],$$

where AD_g is the current action distribution of agent g. The first term describes the immediate self-impact as a consequence of agent g performing action a, while the second term describes the expected immediate impact that the network (i.e., the remaining agents) has on agent g.

Analogously to regular MDPs, a RepNet-MDP agent g strives to maximize its expected discounted perceived impact $\mathbb{E}\big[\sum_{t=0}^{k}\gamma^t PI_{g,t}\big]$, where γ is the *discount factor* and $PI_{g,t}$ is agent g's perceived immediate impact at time-step t. This is accomplished by computing the optimal value function $V_g : \Theta_g \times \mathbb{N} \to \mathbb{R}$ (in a *finite-horizon* setting). It satisfies the *optimality equations*, which are defined as $(\forall \theta_g \in \Theta_g)$

$$\begin{cases} V_g(\theta_g, k) := \max\limits_{a\in\mathcal{A}} \Big\{ PI_g(s, AD_g, a) + \gamma \sum\limits_{s'\in\mathcal{S}} T_g(g, s, a, s', r_g) V_g(\theta'_g, k-1) \Big\} \\ V_g(\theta_g, 1) := \max\limits_{a\in\mathcal{A}} \Big\{ PI_g(s, AD_g, a) \Big\} \end{cases} \tag{2}$$

where $r_g = REP_g(g, Img_g)$, $\theta_g = \langle s, AD_g, Img_g \rangle$, and
$\theta'_g = \langle s', ADE(g, s', AD_g, Img_g), IE(g, Img_g, \alpha, s, AD_g) \rangle$.

In this work, we implement (approximate) online planning [12] instead of exact planning. The general principle of model-based online planning can be described as the interleaving of two phases, the *planning phase*, in which the (PO)MDP performs a look-ahead search of a given depth D, starting at the current environment state, the goal being to determine the most suitable action, and the *execution phase*, in which this action is applied to the environment [12]. In this paper, we make use of an implementation of this approximate technique for the RepNet-MDP framework. We refer to [9] for details on the implementation.

7 Experiments

The goal of the experiments is to showcase the strengths and shortcomings of the framework. To this end, the experimental setup consists of 2 trading scenarios, for which several experiments are conducted. All experiments were conducted with *look-ahead depth* $D = 3$, *Laplace smoothing parameter* $\eta = 0.1$, and *discount factor* $\gamma = 0.7$. Note that these experiments serve as a proof of concept for the RepNet framework and, as such, are not designed to reflect the framework's applicability to problems of realistic scale.

7.1 Experiment 1: Trading between two agents

Let A and B be two agents. Agent A plays the role of the buyer, agent B the role of the seller. Agent A can engage in a trade with agent B, and B can accept or refuse the trade offer. Furthermore, agent A can, prior to making a trade offer, do a good deed in an effort to improve its image in the eyes of agent B.

In this series of experiments, Agent A is managed by the RepNet algorithm. Agent B is run by a simple algorithm that accepts or rejects trade offers made by

agent A according to a set schedule. In particular, agent B is asked to reject trade offers for the 20 first time-steps, accept them for the 60 subsequent time-steps, and finally reject them for the last 20 time-steps.

Two series of experiments are conducted, the first one without making use of subjective actions, the second one by modeling the action of agent A awaiting agent B's response to a trade offer as a subjective action, meaning the outcome of agent A's planning will be influenced by its reputation. A well-designed subjective transition model, schematized in Fig. 3, that realistically reflects how the reputation of agent A may influence the willingness of agent B to accept A's trade offers is put to the test. The variables tracked are the action distribution, image, and by extension the reputation of both agents *in the eyes of agent* A, and frequency at which agent A makes trade offers.

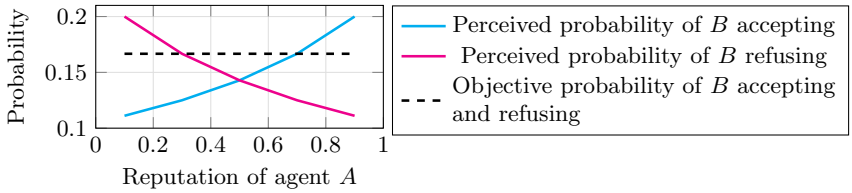


Fig. 3: Perceived probability of agent B accepting and refusing the trade offers, as a function of the self-reputation of agent A.

Fig. 4 shows the evolution of agent A's action distribution for target agent B. Fig. 5 shows the evolution of agent A's self-reputation during the experiment involving the subjective transition model. Note that A's self-reputation, and more generally A's image function, have no bearing on its decision-making if no subjective actions are used (see Equations 1 and 2, T_g makes use of the notion of reputation only for subjective actions). Finally, Fig. 6 shows the evolution of the frequency at which A makes trade offers.

In the first 20 time-steps, B refuses each trade offer. Regardless of the series of experiments, agent A is able to pick up on this via the action distribution. As a consequence, it quickly reduces the frequency at which it attempts to trade with B. In the 60 following time-steps, B is asked to change its behavior and accept each trade offer. Hesitant at first, A gradually increases the frequency at which it attempts to trade with B. Agent A is able to pick up on B reverting back to its old behavior during the final 20 steps.

Additionally making use of a well-designed subjective transition model noticeably improves the RepNet agent's performance. While the trajectories showcase the same key elements, the pace at which agent A is able to adapt improves greatly. The subjective transition model was designed such that agent A believes that its reputation must be good for B to be willing to trade with A (Fig. 3). As such, during the first 20 time-steps, A's relatively poor-in-comparison self-reputation has an immediate negative effect on the *value* it associates with the `trade_with_B` action during the look-ahead search. It quickly becomes more *valuable* to stop trading with B. Similarly, A's reputation needs to be high for it to start trading with B again, explaining the slow increase of the frequency of trade offers at the start of the second phase.

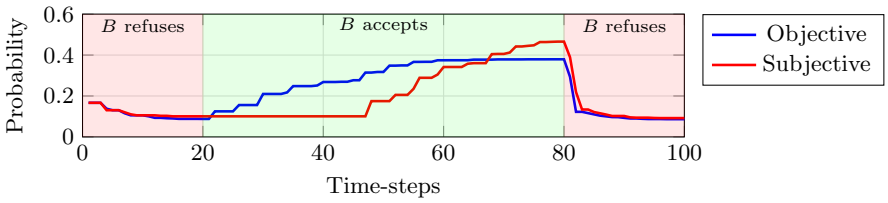


Fig. 4: Probability of B accepting A's trade offers, according to A

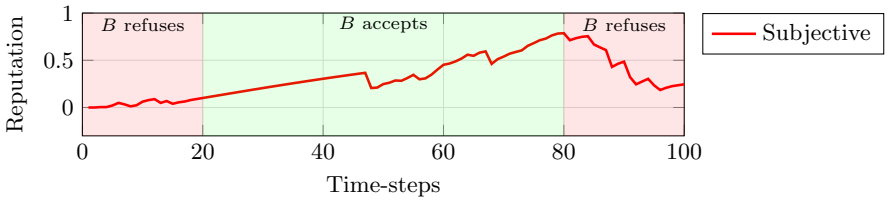


Fig. 5: Reputation of agent A, according to itself.

7.2 Experiment 2: Trading between three agents

Let A, B, and C be three agents. Each agent simultaneously plays the role of buyer and seller, and can thus engage in a trade with any other agent. Each agent can accept or refuse any trade offer made by any remaining agent.

The present scenario is used to verify the ability of a RepNet agent, say agent A, to manage its trades with the two remaining agents B and C, based not only on their behavior towards the agent of interest but also their behavior with each other.

In the first part, agent B is asked to refuse each trade offer made by agent A, while agent C is expected to accept each trade offer coming from A. This portion of the experiments assesses the ability of the RepNet agent (agent A) to accurately determine which agent it is more likely to successfully engage in trades with. In the second part, the roles are switched, and agent B accepts the trade offers, while agent C refuses them. This portion assesses the ability of the agent of interest to *unlearn* what it has learned and adapt its behavior accordingly. In the third and final part, the RepNet agent is asked to not trade with either B or C, that is, to only make use of the `wait` action. Said differently, the optimal action according to its planning, while tracked throughout the experiment, is not performed on the environment. All the while, agents B and C are asked to

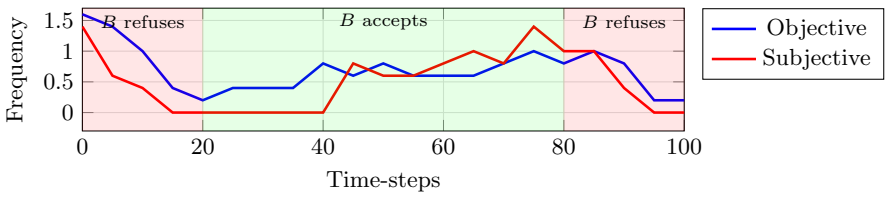


Fig. 6: Frequency of the trade offers made by A, measured in 5 time-step intervals

engage in trades with each other. Agent B is asked to reject all trade offers, while agent C is asked to accept all trade offers. The variables tracked are the action distribution and reputation of B and C *in the eyes of agent* A, as well as the evolution of whom agent A would rather trade with. This portion of the experiment aims at testing the ability of the RepNet agent to draw conclusions on how it should act based on interactions it is not directly affected by.

Fig. 7 shows the evolution of the reputations of agents B and C. Fig. 8 displays the evolution of the probabilities of agents B and C accepting trade offers from agent A. Finally, Fig. 9 shows the evolution of whom agent A would rather trade with.

Agent B is told to refuse, and agent C to accept, each trade offer during the first 33 time-steps. In accordance with the results obtained in Section 7.1, agent A is able to pick up on the other agents' behavioral habits it is affected by. As a result, the reputation of B and its probability of accepting trade offers decrease. Similarly, the reputation of C and its probability of accepting trade offers increase. All the while, agent A chooses to conduct the majority of its trades with C. The following 33 time-steps reverse B's and C's roles. Similarly, agent A is able to adapt its behavior accordingly and ends up trading mostly with B. The reputation of B has increased, while the reputation of C has decreased.

During the last 33 time-steps, agents B and C are tasked with trading with one another while A plays the role of observer, that is, only makes use of the `wait` action. B is asked to refuse all trade offers, while C is asked to accept all trade offers. Interestingly, Fig. 9 shows that, based on its planning, agent A would prefer to keep trading with B, even though the reputation of B decreases and the reputation of C increases in the eyes of A. Said differently, as long as B does not refuse A's offers directly, agent A will prefer to trade with B over C.

The explanation for this is twofold. Firstly, the subjective transition probability of a trade A might want to do with B is, in the eyes of A, conditioned only by A's *own* reputation. As such, B's falling or rising reputation has no bearing on A's decision-making. Secondly, the probability of B accepting (or refusing) A's trade offer, according to A, can only be updated through the direct experience it has with B. As such, the action distribution does not change and can thus not influence A decision-making either.

The simplest way of alleviating this shortcoming is to extend the subjective transition model. Adding the reputation of the agent at the receiving end of the trade offer (e.g., agent B) as a parameter to the subjective transition model would allow agent A to incorporate other agents' reputation in its decision-making process. As such, if the subjective transition probability of B accepting A's trade offer were given by $ST_A(A, \texttt{offer\_state}, \texttt{wait\_s}, \texttt{accept\_state}, r_A, r_B)$, where the newly introduced parameter r_B is B's reputation, agent A could make use of r_B to assist with its decision-making. This comes with the drawback of increasing the complexity of designing the subjective transition model.

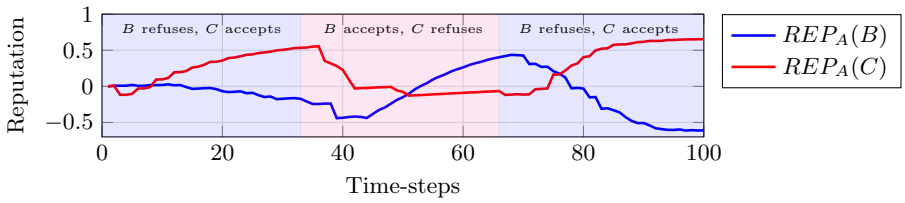


Fig. 7: Reputation of agents B and C, according to A

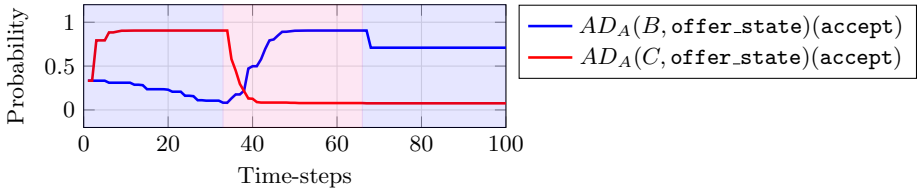


Fig. 8: Probability of agents B and C accepting trade offers from agent A, according to A

8 Summary and future work

In this paper, we revised the multi-agent framework called RepNet introduced by *Rens et al.* [11], addressed its mathematical inconsistencies and proposed a online learning algorithm for finding approximate solutions. The viability of the framework was then tested in a series of experiments.

The current definition of *objective* transitions could be extended to incorporate the reputation of agents other than the RepNet agent. The experimental results showed that the RepNet agent is incapable of adapting its behavior to situations that do not directly affect it. Including the reputation of the agent at the receiving end of a directed action in the *directed* transition model is likely to lead to better-informed decision-making.

We did not address partially observable environments. Many real-world problems do not benefit from full observability, bringing the updated RepNet framework back to a partially observable setting should be considered for future work.

The small-scale experiments conducted in Section 7 served as a proof of concept for the RepNet framework. While applying the framework to problems

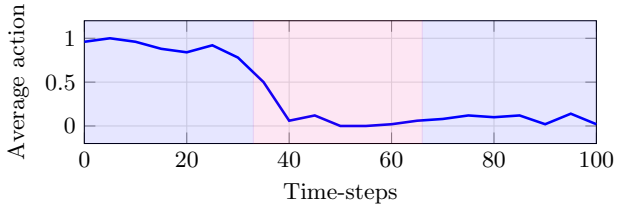


Fig. 9: Average action taken by agent A. Action $a = 0$ corresponds to trading with agent B, action $a = 1$ corresponds to trading with agent C.

of realistic size was beyond the scope of this paper, the absence of large-scale tests does raise questions as to the scalability of this approach. Real-world problems can easily become too complex for transition models to be designed by any one person without leveraging common state features [5]. A compact way to represent real-world state spaces can be achieved by introducing elements of relational logic [5]. From a *logic programming* point of view, a state space is hereby defined by a collection of relations, while a state is an *interpretation* of this collection [8]. Transition models and reward schemes are then represented by *probabilistic rules* [10].

References

1. Abbeel, P.: Learning for Robotics and Control - Value Iteration, CS294-40, University of California, Berkeley (2008), https://inst.eecs.berkeley.edu/ cs294-40/fa08/scribes/lecture2.pdf
2. Becker, R., Zilberstein, S., Lesser, V., Goldman, C.: Solving Transition Independent Decentralized Markov Decision Processes. J. Artif. Intell. Res. (JAIR) **22**, 423–455 (07 2004)
3. Bernstein, D.S., Zilberstein, S., Immerman, N.: The Complexity of Decentralized Control of Markov Decision Processes. CoRR **abs/1301.3836** (2013)
4. Boutilier, C.: Planning, Learning and Coordination in Multiagent Decision Processes. In: Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge. pp. 195–210. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1996)
5. Boutilier, C., Reiter, R., Price, B.: Symbolic Dynamic Programming for First-Order MDPs. pp. 690–700 (01 2001)
6. Busoniu, L., Babuska, R., De Schutter, B.: A Comprehensive Survey of Multiagent Reinforcement Learning. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **38**, 156 – 172 (04 2008)
7. Doshi, P., Gmytrasiewicz, P.J.: A Framework for Sequential Planning in Multi-Agent Settings. CoRR **abs/1109.2135** (2011)
8. Joshi, S., Kersting, K., Khardon, R.: Self-taught decision theoretic planning with first order decision diagrams. pp. 89–96 (01 2010)
9. Maoujoud, D., Rens, G.: Reputation-driven Decision-making in Networks of Stochastic Agents (2020), https://arxiv.org/abs/2008.11791
10. Nitti, D., Belle, V., De Laet, T., De Raedt, L.: Planning in hybrid relational MDPs. Machine Learning **106**(12), 1905–1932 (Dec 2017)
11. Rens, G., Nayak, A., Meyer, T.: Maximizing Expected Impact in an Agent Reputation Network - Technical Report. CoRR **abs/1805.05230** (2018), http://arxiv.org/abs/1805.05230
12. Ross, S., Pineau, J., Paquet, S., Chaib-draa, B.: Online Planning Algorithms for POMDPs. CoRR **abs/1401.3436** (2014)
13. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edn. (2009)
14. Wiering, M., Otterlo, M.: Reinforcement Learning: State-Of-The-Art, vol. 12 (01 2012)

Learning to Classify Users in the Buyer Modalities Framework to Improve CTR*

Laurent Mertens[1,2], Peter Coopmans[3], and Joost Vennekens[1,2]

[1] KU Leuven, De Nayer Campus, Dept. of Computer Science
J.-P. De Nayerlaan 5, 2860 Sint-Katelijne-Waver, Belgium
firstname.lastname@kuleuven.be
[2] Leuven.AI - KU Leuven Institute for AI, B-3000 Leuven, Belgium
[3] Addrelevance
Grevensmolenweg 28, 3800 Sint-Truiden, Belgium
peter.coopmans@addrelevance.be

Abstract. The Buyer Modalities framework divides buyers into 4 profiles, where each profile has its own specifics as to how it makes its purchasing decisions. We built an online prediction system that categorizes website visitors based on this framework. According to this categorization, a specific banner ad variant tailored to that profile was shown to the visitor, rather than a default "neutral" variant, resulting in a significantly improved CTR.

Keywords: Data mining · Predictive modeling · Ensemble methods · Online advertising.

1 Introduction

The Buyer Modalities framework [3] is a model that defines four distinct psychological profiles of consumers according to how they make their purchasing decisions. These four types – competitive, spontaneous, methodical and humanistic – are illustrated in Fig. 1, and are based on two main axes: decision speed (impulsive vs. deliberate) and rationale (emotional vs logical). It states that each profile reacts to different types of information. If we consider, e.g., the purchase of a new car, people with a methodical profile will be more interested in a detailed list of features of the car as can be found in the brochure, whilst the humanistic profile will be more served with testimonials from people who already own the car.

The implication of this model for advertising is that in order to have an effective campaign, ideally each profile is targeted with an ad tailored toward its information needs. The issue with this of course is that one needs to know the profile of the user, which one typically does not. In order to remedy this issue, we propose a framework that uses historical user-website interaction data

* Supported by the Flanders Innovation & Entrepreneurship TETRA project "Start To Deep Learn".

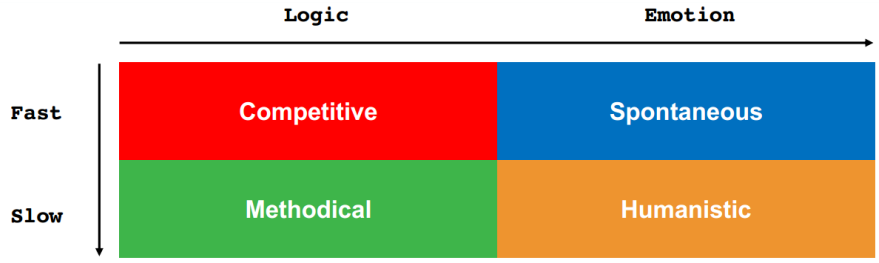


Fig. 1. A schematic overview of the four psychological profiles described by the Buyer Modality framework.

to predict the user profile when needed, and hence allows the ad server to display the appropriate variant for this particular user in a dynamic way. We show that this approach results in a significant increase in the click-through rate (CTR), i.e., the percentage of users viewing a web page who click on an ad displayed on that page.

Ads come in many forms, e.g., pop-ups or "sponsored content", each with its own specifics. Our work involves so-called "banner ads", banner-like graphical ads often displayed at the top or in the margins of a page. Ever since the advent of online advertising, the CTR has served as a key measure for the succes of an ad campaign. Given the multi-billion industry that is online advertising, the question of what makes an ad effective has been given quite some attention.

Specifically for banner ads, [8] look at the effect of the banner ad size, style and orientation on its success. In [1], the authors attempt to predict the CTR, rank according to CTR and categorize into "high" and "low" CTR a set of $\pm$10K banners by using a custom defined set of 43 different visual features. In all three tasks, they manage to consistently outperform the baseline. In contrast, [7] performed two eye tracking studies to investigate the relation between visual design and relatedness to the page content, and visual attention devoted to the ad. In a first study, they show a professionally designed graphical ad to one group of participants and a text-only banner (with the same text as the graphical ad) to another. Besides this, half of the ads (graphical and text-only combined) were related content-wise to the page, whilst the other half were not. They found that none of these parameters had a statistically significant effect on the dwell time. This prompted a second study, in which they showed that dwell time does increase significantly if an ad is relevant to a user's intent or task, rather than to a page's content. Somewhat closer ideologically to our work is [4], who studied the effect of demographically targeting banner ads on users' visual attention and brand evaluation. This kind of targeting focuses on demographic properties of the users such as gender, age and location, and follows the assumption that "similar people act in a similar way". Hence, by tailoring ads to these properties, it should be possible to increase user attention. They found that targeting ads

this way does indeed increase users' visual attention, but not necessarily their brand evaluation.

2 Conceptual setup

In this section we will first provide a conceptual description of our work, followed by a detailed description of the concrete implementation for our use case in the following section.

Conceptually, our setup is the following. Given a particular website that displays ads, we track certain user-website interaction data and use this historical data to extract features to be fed to a predictive model. Ideally, an expert should identify salient elements to be tracked, e.g., specific hyperlinks or other so-called Calls to Action (CTA), that are likely to appeal more to one profile rather than the others.

We distinguish two phases. During a first phase data is being collected to be used to train a predictive model. During this phase, different variations of the same ad targeted to each profile will be displayed at random. This means that a single user can get served different variants of the same ad. When a user clicks on one of the variants, we assign the associated profile to the user to obtain the training targets.

During the second phase, we continue to collect user-website interaction data, and use this data to query the predictive model we obtained in phase one in real time to obtain a user profile. This prediction then determines which ad variant will be shown to this user.

3 Case specifics

Our experiment was performed in collaboration with a commercial partner, Produpress [6], a company that owns amongst others a number of automotive magazines and corresponding websites. We worked with two websites, www.autogids.be (Autogids) and www.moniteurautomobile.be (Moniteur), which are essentially the Dutch and French language versions of the same content. Both sites target a Belgian audience. A large part of the content are extensive car reviews. There were some differences in data collection during training and deployment phases, which we will discuss in the following sections. These differences are due in large part to the fact that data collection was performed by a third party for the first phase, whilst being performed by ourselves for the second phase.

3.1 First phase

In this phase, four different variations of an ad banner for a specific car ad where designed, one for each buyer modality, which were shown on a random basis. The main difference between the variants was the CTA used (i.e., text), rather than

the graphics. If a user clicked on a banner, the profile targeted by the variant becomes the profile of the user. E.g., if a user clicked on the "competitive" variant, when training the model later on we take this user as being a target for the "competitive" profile.

The number of positive samples we collected per profile can be seen in Table 1. Note that some users clicked on more than one ad variant; there were 353 unique users who clicked on a variant for a total of 370 clicks. These users where treated as targets for all the variants they clicked on. The numbers used in this and following tables correspond to the different types as follows: 1 = competitive, 2 = spontaneous, 3 = methodical and 4 = humanistic.

Table 1. Number of collected positive samples per profile

	Autogids	Moniteur	All
1	89	79	168
2	46	30	76
3	38	26	64
4	34	28	62
Total	207	163	370

Besides target labels, also aggregated and custom features for each distinct user were collected. The 7 custom features basically correspond to (the URLs leading to) the main sections of the car reviews, here translated from Dutch: "Read our test report", "View the gallery", "Robotportrait and conclusion", "Tested version", "Users reviews", "Compare this car" and "Find a dealer". With these, the set of features for this specific experiment consists of, per user and over the data collection period (abbreviations correspond to Fig. 2):

- The number of pageviews. (PgV.)
- The average time spent per pageview. (AtP.)
- The number of sessions. (#Ses.)
- The average time spent per session. (ASD.)
- Per custom feature: the number of sessions the user saw this particular content type, i.e., clicked on the corresponding URL. (CT1–CT7)
- Per ad variant: the number of sessions the user was shown this particular ad variant. (AV1–AV4)

Note that at this stage, we did not have any other information besides these aggregated features. This means that we were unable to determine when exactly a user clicked on an ad, which in turn means that all aggregates were determined by also taking into account data from *after* when a user clicked an ad. Ideally, these statistics would have only been determined by using data prior to a click. Fig. 2 shows the average feature values between clicking and non-clicking users for each ad variant separately. This graph clearly illustrates that indeed there appears to be a behavioral difference between both groups of users, as indicated

by the fact that for "click" samples the average values are consistently higher than for "no click" samples.

To further analyze the data, we first checked whether or not we could distinguish between "click" and "no click" samples in general, regardless of profile type. For the remainder, all models were trained using Python's Scikit-learn package [5]. Table 2 contains the average accuracy over 200 Random Forest classifiers, each consisting of 200 trees with `max_depth = 3`. For each iteration, a random selection of negative samples was chosen to complement the positive ones, and 20% of the data was held out as test data. As the data shows, performance is far better than random, although also far from perfect, with Autogids and Moniteur performing very similarly, suggesting that it is indeed possible to predict what users are more inclined to click on an ad, regardless of profile type.

Table 2. Click vs. No Click classification performance over all ads by means of a Random Forest classifier. Average performance over 200 forests of depth 3.

Dataset	Train	Test
Autogids	0.743 ± 0.017	0.679 ± 0.054
Moniteur	0.738 ± 0.018	0.655 ± 0.062

In a next step, we checked to what extend it was possible to distinguish between each pair of profiles. The assumption is that if there is no correlation between user profiles and ad variants, users will randomly click an ad variant and hence it will not be possible to discriminate between ad pairs. To test this hypothesis, we again looked at the average accuracy over 200 Random Forest classifiers with 200 trees each and `max_depth = 3`, with an 80/20 train/test data split. The results are shown in Table 3, and except for the last pair (3 vs 4) show performance that is in line with the "click" vs "no click" scenario. This indicates that it is possible, up to a point, to discriminate users based on the ad variant they clicked. In other words: different people do have a different ad variant preference.

Table 3. Full dataset: Random Forest accuracy per ad pair.

Ad Pair	Train	Test
1 vs 2	0.767	0.650
1 vs 3	0.836	0.689
1 vs 4	0.852	0.685
2 vs 3	0.854	0.576
2 vs 4	0.857	0.612
3 vs 4	0.806	0.501

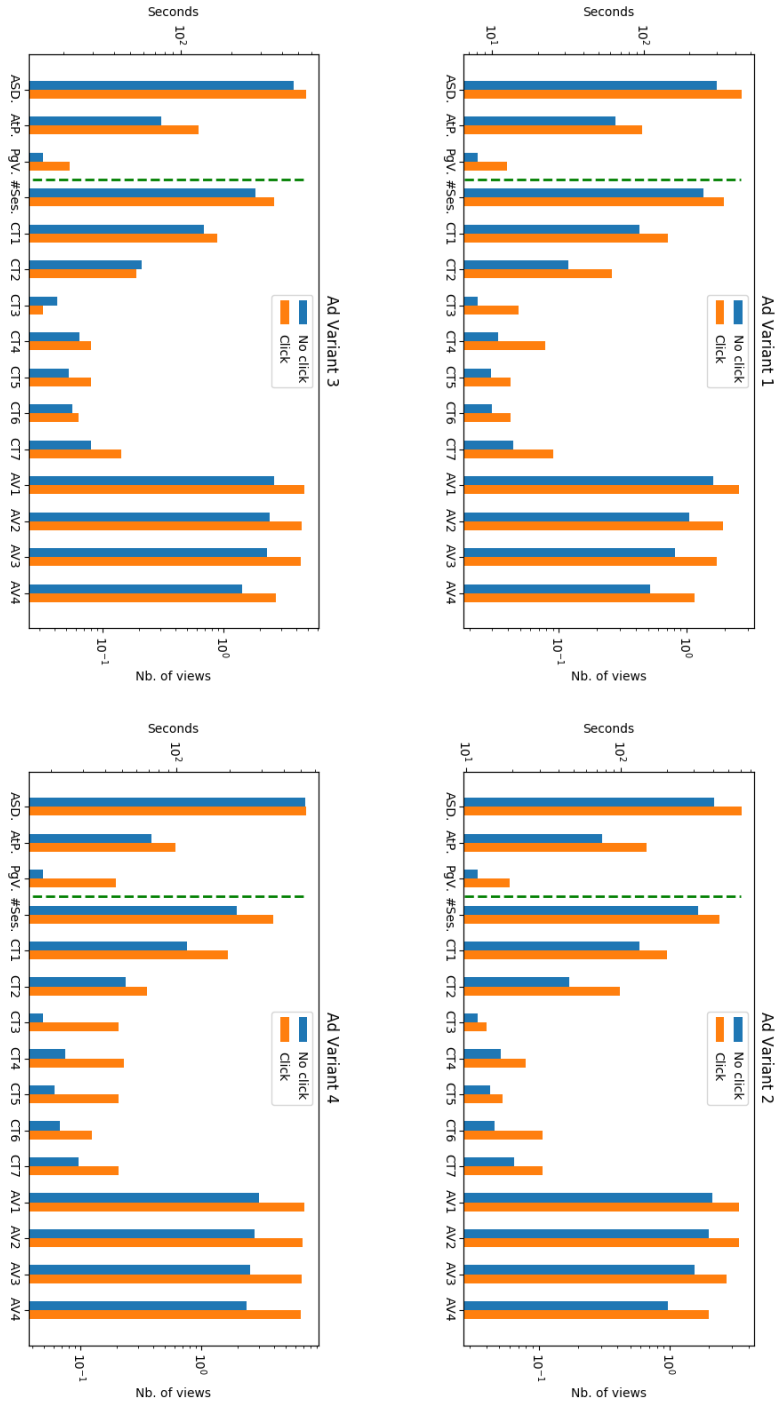


Fig. 2. Averages per feature between "click" and "no click" samples for each ad variant.

Since our assumption is that each user, regardless of whether they click an ad or not, can be described by one of the four buyer modalities, we wish to train a model that always predicts one of these profiles, and does not make a "neutral" prediction. As training data we used all positive samples of all four variants combined (i.e., no negatives), and again opted for a Random Forest model with 200 trees, albeit multiclass this time, to predict one of the four profiles for each user. Train and test sets were stratified so as to have equal ratios of samples per class. Table 4 shows results with `max_depth = 3` and `max_depth = 10` settings by again taking the average over 200 iterations, with an 80/20 data split at each iteration.

Table 4. Full dataset: multiclass Random Forest accuracy

`max_depth`	Train	Test
3	0.515 ± 0.013	0.470 ± 0.022
10	0.946 ± 0.011	0.405 ± 0.043

Performance is considerably lower than with previous experiments, even taking into account the fact that the baseline is 0.25 this time. As can be expected, the experiment with `max_depth = 10` results in overfitting, as apparent by the large discrepancy between train and test accuracies. Nevertheless, we chose to go with this model for phase 2, as our philosophy was that given the low number of samples at our disposal, we preferred the model to overfit on these so that they can serve as stringent prototypes, rather than making a more "diffuse" model.

3.2 Second phase

The second phase was ran in light of a specific advertisement campaign for a new car. The compaign ran for four weeks total; two on Autogids and two on Moniteur. Similar to phase one, five variants of the ad banner were made: one for each profile, plus a "neutral" variant in case the profile of the user could not adequately be predicted. A major difference with the first phase is that we collected the user-website interaction data ourselves. This allowed us to compute the features in an online way. Data was collected by means of a custom JavaScript script, that would send the data to a PHP service to be stored in a MySQL database. Information stored included, a.o., a unique user ID, page visits and clicks.

To allow the website to request a profile for a visiting user, we developed a Python API using CherryPy [2]. Whenever a profile was requested, the known data for this user would be retrieved from the SQL database, and the same features as used during phase one computed on the fly. A prediction was only made if the user had visited the site during at least 3 sessions (including the one at prediction time). If this was the case, we would then verify that the highest profile score returned by the model > 0.35. If so, the corresponding profile would

be returned. In all other cases (also if the user ID was unknown), a default "neutral" profile would be returned. Recall that we used the `max_depth = 10` model described at the end of §3.1, whose performance is shown in Table 4.

We are not allowed to report specific CTR numbers because of contractual obligations to our commercial partners. Hence, we can only report changes w.r.t. the baseline. For both Autogids and Moniteur, a baseline CTR was determined over 14909 and 12502 impressions of the "neutral" banner respectively. This means that the users that belonged to this control group did net get to see a banner based on our predictive system. The CTR for our system was determined over 63284 and 56751 impressions respectively. This does not mean that all users belonging to the test group saw a customized banner, simply that for these users, we attempted to make a prediction. The CTR on banners displayed using our system were 31% and 35% higher than the baseline CTR for Autogids and Moniteur respectively, for an average increase of 33%.

Given this result, it was decided to run a second campaign, for a different car by the same brand as the first campaign, using our system over a period of four weeks, but without further involvement from our part. CTR for this campaign were 129% and 94% higher that the baseline determined in the previous campaign. Unfortunately, a new baseline was not determined and hence these numbers are only reported by way of indication.

4 Conclusion

In this work, we described how the Buyer Modalities framework can be used to improve the CTR on online ads. We built a Random Forest model based on features extracted from aggregated web analytics, and used this model in a system that allows to predict the Buyer Modality profile of a website visitor. Using this predicted profile to dynamically adapt the ads shown to the user resulted in a 33% improvement in CTR compared to the reference user group.

We would like to point out that our method theoretically does not require a new training phase for each new ad, since although the ads change, the user modality profiles do not. This implies that, given careful design of the ad variants, once a model has been trained it should be applicable to any ad campaign. Consequently, by collecting data over several campaigns, the model can also continually be further improved by incrementally retraining the model.

Moreover, for this particular experiment the raw data consisted of aggregated features. We expect that having data available at a more granular level, as collected by ourselves in phase 2, should allow the development of more and better features to further improve the accuracy of the predictive model.

References

1. Azimi, J., Zhang, R., Zhou, Y., Navalpakkam, V., Mao, J., Fern, X.: Visual appearance of display ads and its effect on click through rate. In: Proceedings of

the 21st ACM International Conference on Information and Knowledge Management. p. 495504. CIKM '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2396761.2396826, https://doi.org/10.1145/2396761.2396826
2. CherryPy: https://cherrypy.org/
3. Eisenberg, B., Eisenberg, J.: Waiting for Your Cat to Bark. Thomas Nelson (2006)
4. Kaspar, K., Weber, S.L., Wilbers, A.K.: Personally relevant online advertisements: Effects of demographic targeting on visual attention and brand evaluation. PloS one **14**(2), e0212419–e0212419 (2019)
5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
6. Produpress: https://www.produpress.be/
7. Resnick, M.L., Albert, W.: The influences of design esthetic, site relevancy and task relevancy on attention to banner advertising. Interacting with Computers **28**(5), 680–694 (2016)
8. Sigel, A., Braun, G., Sena, M.: The impact of banner ad styles on interaction and click-through rates. Issues in information systems **9**(2), 337–342 (2008)

Comparing Exploration Approaches in Deep Reinforcement Learning for Traffic Light Control

Yaniv Oren[1], Rolf A. N. Starre[1], and Frans A. Oliehoek[1]

Technical University Delft, Mekelweg 5 Delft, the Netherlands {y.oren@student., r.a.n.starre@, f.a.oliehoek@}tudelft.nl

Abstract. Identifying the most efficient exploration approach for deep reinforcement learning in traffic light control is not a trivial task, and can be a critical step in the development of reinforcement learning solutions that can effectively reduce traffic congestion. It is common to use baseline dithering methods such as ϵ-greedy. However, the value of more evolved exploration approaches in this setting has not yet been determined. This paper addresses this concern by comparing the performance of the popular deep Q-learning algorithm using one baseline and two state of the art exploration approaches, and their combination. Specifically, ϵ-greedy is used as a baseline, and compared to the exploration approaches Bootstrapped DQN, randomized prior functions, and their combination. This is done in three different traffic scenarios, capturing different traffic profiles. The results obtained suggest that the higher the complexity of the traffic scenario, and the larger the size of the observation space of the agent, the larger the gain from efficient exploration. This is illustrated by the improved performance observed in the agents using efficient exploration and enjoying a larger observation space in the complex traffic scenarios.

Keywords: Reinforcement learning · Traffic optimization · Exploration.

1 Introduction

Traffic congestion is a global predicament. For instance, in the EU alone its cost is estimated to be 1% of the EU's GDP [15]. One approach for reducing this cost is optimization of traffic flows by improving traffic light control policies. To find such policies, reinforcement learning (RL) has a strong appeal as a paradigm that is able to find high performance solutions to sophisticated problems. Research has been done into the application of RL to the problem of traffic light control optimization in the past [1],[4],[13],[14],[18], often specifically concerning application of deep RL algorithms [4],[13],[18].

A fundamental principle of RL is exploration, and the balance between exploration and exploitation. Namely, how much does the agent explore its environment, versus how much it opts for actions that it expects to return the most cumulative reward. Many different exploration approaches exist for reinforcement learning [3],[6],[10]–[12]. These approaches often perform differently in different settings, in addition to having different computational costs [11],[12]. It has been shown that for some RL settings, simple exploration approaches such as ϵ-greedy are insufficient for RL to be able to perform well, or at all [11],[12], which can be caused by the reward function used and the complexity of the specific problem tackled. This illustrates the importance of identifying effective exploration techniques for specific RL settings. To the best of our knowledge, there has not been an attempt to investigate the importance of efficient exploration in deep RL in the setting of traffic light control.

This paper investigates a comparison between different exploration approaches in deep RL for traffic light control. This, to facilitate better deep RL by identifying the value of evolved exploration approaches in this setting, such as higher sample efficiency, or higher final policy score. For that purpose, this paper compares the performance of the popular deep Q-learning algorithm (DQN) [10] using one baseline and two state of the art exploration approaches, and their combination. Specifically, ϵ-greedy is used as a baseline, and compared to the exploration approaches Bootstrapped DQN, randomized prior functions, and their combination. This is done in three different traffic scenarios, ranging from simplified to simulating real traffic, in order to investigate the effect of exploration in different traffic profiles.

This paper first introduces a theoretical background for deep RL and exploration. Second, a description of the exploration techniques compared, along with the modeling of traffic light control as an RL problem, are given. This is then followed by an explanation of the research methodology and the experimental setup, leading to a presentation of the results obtained and their analysis. Last, ethical and epistemic concerns are considered, implications of the work are discussed and conclusions are drawn.

Altogether, the results obtained suggest a link between the complexity of the traffic scenario, the amount of information accessible to the agent, and the gain from efficient exploration. This is illustrated by the improved performance observed in the agents using efficient exploration and enjoying a large observation space, in the complex traffic scenarios.

2 Background

This section introduces background information relevant to the work presented in this paper. First, an overview of reinforcement learning (RL), is given, laying the basis for an introduction to deep RL and a description of the deep Q-learning (DQN) algorithm [10] that follows. Last, the principle of exploration is explained, followed by an overview of dithering [12], deep and directed exploration.

2.1 Reinforcement learning

In RL, an agent operates in an environment. The environment provides information regarding the state the agent is in and what actions it can execute. The environment is usually described in the form of a Markov decision process (MDP), a stochastic control process often used to model decision making in partially stochastic domains. A Markov decision process is represented as a four-tuple, $M = (S, A, P, R)$, where S represents the state space, A the action space, P the transition function and R the reward function.

The agent interacts with the environment by observing a state $s_t \in S$, executing an action $a_t \in A$, and receiving a reward $r_t \in R$ for the action executed. The agent is attempting to learn a policy π, such that the expected reward over time is maximised.

In the Q-learning algorithm [17], the value of state-action pairs is estimated by the agent, using iterative Bellman updates: $Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[y_t - Q_t(s_t, a_t)]$, where α is the learning rate, and the target $y_t = r_t + \gamma \max_a Q_t(s_{t+1}, a)$. s_{t+1} denotes the new state arrived at after choosing action a_t in state s_t, a any action available at state s_{t+1}, and $0 \leq \gamma \leq 1$, is a discount factor. In many scenarios however, the state-action pair space is too large for the computation or memorisation of each value $Q(s, a)$ to be tractable. To avoid this problem, function estimators such as neural networks can be used to estimate the Q value. This gives rise to the use of neural networks in reinforcement learning, and specifically the deep Q-Learning (or deep Q-networks) algorithm, commonly referred to as DQN [7], [10].

2.2 Deep reinforcement learning with DQN

The DQN algorithm uses deep neural networks to estimate a mapping from states to Q-values [10]. Instead of saving or computing each $Q(s, a)$ value separately, the algorithm learns a parameterized value function $Q(s, a; \theta_t)$. As a result, rather than learning the Q-values directly, the algorithm learns the parameter set θ of the Q-function. The previous Q-learning update then becomes:

$$\theta_{t+1} = \theta_t + \alpha(y_t - Q(s_t, a_t; \theta_t))\nabla_{\theta_t} Q(s_t, a_t; \theta_t)$$

Here, $y_t = r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a; \theta_t)$. To prevent instabilities, the DQN algorithm uses an additional network, termed the target network, θ^- [10]. The target network is the same as the regular, or online, network. However, it only updates every certain τ time-steps, by copying the parameters θ of the online network. This target network is used by the DQN algorithm in the target term, which becomes instead: $y_t = r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a; \theta_t^-)$.

Double DQN [16] is a commonly used modification suggested to the original DQN algorithm, which aims to reduce overoptimism caused by estimation errors, which DQN is prone to. This is done by decoupling the selection of an action from its evaluation [16]. In vanilla DQN, in the term $y_t = r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a; \theta_t^-)$, the agent uses the same network θ^- for both selecting and evaluating an action. The double-DQN algorithm proposes using the online network θ to choose the action, and the target network θ^- to evaluate the choice. The target term y_t used in the double DQN update then becomes:

$$y_t = r_{t+1} + \gamma Q_t(s_{t+1}, \arg\max_a Q_t(s_{t+1}, a; \theta_t); \theta_t^-)$$

2.3 Exploration in reinforcement and deep reinforcement learning

In order to find an optimal policy through experience alone, which is the general premise of RL, the agent must encounter the rewards that are part of an optimal policy at least once. This leads directly to a necessity to explore the environment - if the agent does not explore, how will it encounter valuable rewards that do not lie over its existing policy's path? However, the agent is also expected to efficiently converge into an optimal policy, and not only explore its environment. This leads to one of the fundamental principles of RL - exploration vs exploitation. Different approaches have been developed - ranging from dithering, random action choosing exploration [10], to more evolved notions such as deep and directed exploration [11], [12], and others. These approaches each attempt to achieve efficient exploration through different means - from simplicity of computation to effective analysis of the agent's knowledge and uncertainty.

Dithering exploration The common baseline exploration strategy used in DQN is a dithering exploration method, or ϵ-greedy. In ϵ-greedy, the agent takes a random action (i.e., explores) with probability ϵ, and with probability $1 - \epsilon$ the agent takes the best action according to its current Q value estimation. ϵ-greedy achieves state of the art performance against many popular benchmarks [12], [16]. However, as discussed in [12], in environments where rewards are scarce and distanced in the state-action space, and their values have a large spread, the dithering exploration of ϵ-greedy can take an exponentially long time to arrive at high-valued rewards. This raises the necessity for a more advanced type of exploration, that can be directed over over multiple time steps. These concepts have been coined 'directed exploration' and 'deep exploration' [12].

Deep & directed exploration Directed exploration attempts to improve the efficiency of the agent's exploration by directing it. For example, to previously unexplored or under-explored areas of the state-action space. To achieve a measure of directed exploration, an uncertainty measure can be used - the more uncertain the agent is about the value of some state or action, the more it can prioritize exploration. For directed exploration to be effective, it often doesn't suffice for it to be directed over one, or small number of time steps [12], but must be directed over multiple time steps. The term 'deep exploration' is used to describe such an exploration approach, that is directed over multiple time steps [12].

3 Exploration In Deep Reinforcement Learning for Traffic Light Control

This section will first motivate the choice to evaluate exploration techniques that focus on deep and directed exploration. Following that, the agent used in the experiments is outlined, and the exploration approaches it implements are described. Last, the modelling of traffic light control as an RL problem used in this paper is presented.

3.1 Motivation

This paper opts to specifically identify the value of exploration approaches that focus on achieving deep exploration, in the setting of traffic light control. These approaches have been chosen not only for being state of the art in this field, but also for their potential in this setting. In heavy traffic scenarios, suboptimal actions may carry a long term effect. They may immediately cause congestion, and once there, it may be difficult to return to less congested states [13]. Deep exploration may be able to improve the agent's ability to escape such scenarios, by directing its exploration along a specific path. While this path will not necessarily pay in the short run, it may allow the agent to recover from congestion in the longer run. Additionally, the ability of the agent to more efficiently explore areas of the state action space that lie beyond areas plagued by negative rewards, as a result of employing deep exploration, may allow the agent to learn optimal policies that will otherwise be unlikely for a dithering agent to ever achieve.

3.2 The agent

The agent used in the experiments presented in this paper is a DQN agent, using the double-DQN [16] modification. The agent implements the following exploration mechanisms: ϵ-greedy, Bootstrapped DQN (BDQN) [12] and randomized prior functions [11]. The implementation used in this paper, based on [8] and modified for the setting of traffic light control, allows the agent to use any combination of the three different mechanisms listed above. ϵ-greedy has been described in section 2.3, and has been chosen as it is the common baseline exploration used in DQN [10], [16]. BDQN and randomized prior functions have been chosen as state of the art exploration approaches that aim to achieve efficient deep exploration, and in addition, for their ability to elegantly combine for even better exploration. BDQN and randomized prior functions are described below.

Bootstrapped DQN BDQN has been developed in an attempt to achieve a measure of deep exploration, discussed in section 2.3. In order to achieve deep exploration, BDQN approximates a distribution over Q-values, using a bootstrap. Bootstrapping is a technique used to approximate a population distribution from a sample distribution, using random sampling with replacement [5].

The bootstrap can be implemented efficiently using a shared neural network with several heads. The shared network's role is to learn a feature representation, while each head is providing an independent Q-values estimation. A visualization can be found in figure 1. In learning, the algorithm randomly samples an estimator (a 'head') out of the bootstrap, and follows the policy which is optimal for that estimator for some number of steps greedily or ϵ-greedily. In the experiments done in this paper ϵ-greedy is used. The resulting experiences are gathered in a buffer, and are available for all estimators to learn from, under some probability that decides which experiences will be available to which estimator. Each estimator is trained against its own target network / target network head.

In evaluation, an ensemble voting policy is used to evaluate which action has been chosen by most heads. If there is no majority vote, an arbitrary choice is made between the actions chosen by the most heads. The action is then chosen and executed.

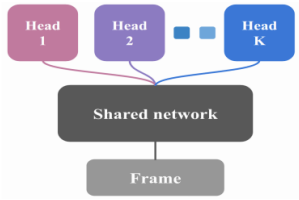


Fig. 1: The BDQN architecture proposed in [12].

BDQN attempts to achieve a measure of deep exploration by following the policy of one of the estimators for some number of steps. For this to be effective, the agent must guarantee that in areas of uncertainty (under-explored areas of the state action space), the different estimators will have different estimations. However, in BDQN this uncertainty, or variety in the Q-value estimations, is only based on the observed data [11]. This can be problematic, because in environments where rewards are very scarce, the agent may learn to believe that there is no reward, and lose all uncertainty, rather than direct its exploration to remote, unexplored areas of the state action space in the hope that they may contain rewards. Such 'prior' drive for exploration, that is independent from the data, is proposed in [11] in the form of randomized prior functions.

Randomized prior functions While usable with a regular DQN agent, the randomized prior functions algorithm is designed to be combined with the BDQN model. To achieve independent uncertainty, the randomized prior functions model consists of one additional neural network for each Q-value estimator, or one shared neural network with one head for each estimator. This additional network or head p is combined with the original estimator f to form the final output Q, through a scaling factor β: $Q = f + \beta p$ [11]. Q is then used in the learning process to minimize the training loss. This results in uncertainty that

is independent from the data: the Q value estimation always includes a neural network initialized with random parameters. No matter the uniformity of experiences the agent encounters (for example, only similar, negative rewards), it will still consider some additional prior 'assumption', in the form of the prior function, in regard to previously un-encountered states. As a result, each estimator will always approximate the Q value of as yet un-encountered states differently. This allows the agent to better direct its exploration, by guaranteeing diversity between the estimations of the different bootstrap heads for previously un-encountered states.

3.3 The model

The modeling of the traffic light control problem as an RL problem used in this paper is follows work done in [13]. The problem is modelled as an MDP $M = (S, A, P, R)$, where S is the state space, A is the action space, P the transition function and R the reward function. The open source traffic simulator SUMO [9] is used to generate the environment.

State space, action space & transition function The state provided to the agent is represented as a set of stacked frames of size $x \in \mathbb{Z}^+$. Each frame is a matrix containing current locations of vehicles in the agent's observation space, and the current traffic light configuration. The observation space of the agent is a square centered at the intersection controlled. Each location of a vehicle is marked with a 1, and empty locations with a 0. The traffic lights configuration is presented in the matrix as numbers between 0 and 1 chosen arbitrarily. Stacking several frames allows the agent to extrapolate vehicle speed from the state representation. $x = 4$ was used in the experiments, to achieve a balance between too much information (complicating the learning process), and too little information (hindering the capacity of the agent to learn effective policies).

The actions available to the agent at each time step are one of two traffic light configurations, representing which lanes receive a green light. The transition function is defined by SUMO.

Reward function The reward function used is a modified version of the reward function developed in [13]. At each time-step t, the agent receives a reward r_t, computed by iterating over all vehicles currently in the agent's observation space, and summing different penalties:

$$r_t = -1.5c - 0.2\sum_{i=1}^{N} e_i - 0.3\sum_{i=1}^{N} d_i - 0.3\sum_{i=1}^{N} w_i$$

This, where i represents the vehicle index. c is a penalty for switching the light configuration, to prevent flickering. e_i is a penalty for sharp decelerations, to penalize emergency stops. d_i is a penalty for the 'delay' of a vehicle, defined as $1 - \frac{vehicle\ speed}{allowed\ speed}$. Finally, w_i is a waiting penalty, defined as 0.5 for the first step of a car standing still, and 1 for any consecutive step.

The modification included removal of a term that punishes teleportation of vehicles (used in SUMO to mark traffic collisions) due to implementation challenges. Additionally, the coefficient of the term c was increased from 0.1 to 1.5, after observation that otherwise the penalty for light switching is barely noticeable with almost any number of cars.

4 Experimental Setup

This section first describes and motivates the methodology used to evaluate the exploration approaches in the setting of traffic light control. This is followed by a description of the three traffic scenarios used to evaluate the exploration approaches.

4.1 Comparison of different exploration methods

To evaluate the impact of exploration in the setting of traffic light control, this paper compares the performance of agents using ϵ-greedy, BDQN [12], randomized prior functions [11] and a combination of the above for exploration, in three different scenarios. The performance is evaluated and averaged over multiple repetitions.

The compared agents As all three exploration approaches investigated are designed to be combined, the performance of the following six agents is compared: a regular DQN agent ad regular DQN agent with a randomized prior function, both employing ϵ-greedy; Two BDQN agents with increasing bootstrap size: 4 & 10 bootstrap samples (or neural network 'heads'); Last, two similar BDQN agents, combining randomized prior functions in their bootstrap mechanism. The number of bootstrap samples has been chosen based on a relation between computational complexity (the larger the bootstrap, the larger the complexity), and gain from the bootstrap (the larger the bootstrap, the better the average performance). As shown empirically in [12], the relative gain from sizes larger than 10 becomes insignificant very quickly. The experimentation with different combinations of those techniques allows to evaluate, in essence, even more exploration techniques, and is the reason it is done in this paper.

The parameters of the agents investigated are not tuned for the specific setting of traffic light control. For the purpose of full reproducibility, a full list of the experiment parameters and agents' hyper-parameters used is available with the code base used in the experiments.

The evaluation The evaluation is done as follows: an experiment is done, for each agent in each scenario and each intersection considered. Each experiment consists of N learning episodes. Every $evaluation\_frequency$ learning episodes, an evaluation phase is ran. In order to reduce sensitivity to stochastic noise from the random nature of the traffic used in the experiments, in the evaluation phase the agent's policy is evaluated over $number\_evaluations$ evaluation episodes, with $\epsilon = 0$. The average episodic reward is then used for evaluation. The exact parameters $evaluation\_frequency$, $number\_evaluations$ used in each experiment are detailed in section 5. To further reduce the impact of stochasticity, the entire experiment is repeated X times for each agent and the results averaged. Finally, the performances of the different agents are plotted against each other, in the form of their averaged evaluations' rewards. The results are presented in section 5.

The different parameters mentioned above were chosen in the following way: N was chosen from experimentation, as the range within which the agents' learning starts to plateau, in order to present the differences between the evaluations of the agents' in the clearest way. The $evaluation\_frequency$ used for each experiment set is chosen to achieve balance between the number of episodes each experiment is ran for, and the number of total evaluation episodes in the experiment. The $number\_evaluations$ parameter has been chosen as a balance between the total number of evaluation phases and the total number of episodes

in each experiment. The larger the number of learning episodes, and lower the evaluation frequency, the larger the *number_evaluations* parameter. To balance computational costs and time constraints with reliability of results, each experiment was chosen to be averaged over $X = 25$ repetitions.

For the purpose of full reproducibility, all random generators used are fully seeded, and the seeds logged. Each experiment is initiated with a different random seed, to guarantee random initialization of the agents' neural networks' weights. The environment's traffic generator however is seeded with the same set of seeds for all experiments. This is done to guarantee that while all agents experimented with are different, they are tested against the same traffic simulations.

4.2 The traffic scenarios

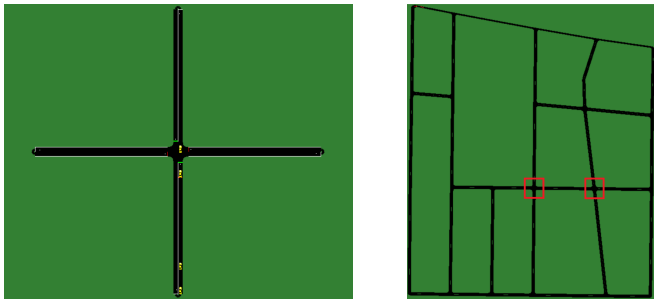
The impact of the exploration approaches investigated in this paper is evaluated using the traffic simulator SUMO [9], in three different traffic scenarios. In the first scenario, one set of experiments is done. In the second and third scenarios, two separate experiment sets are done, evaluating the agent against traffic of slower and faster average speed.

Scenario 1: The grid The first scenario is a basic grid-like road network, with one intersection in the center, and four roads going one in each direction from the intersection: north, east, south and west. A visualization of the grid scenario is presented in figure 2 a. The first scenario is meant to capture a simple, independent intersection profile, that does not consider or experience the behavior of other neighboring intersections. The traffic in this scenario is generated randomly, based on a set number of vehicles over a set spawning time.

Scenario 2: Simulation of real traffic in Manhattan, New York The second scenario, visualized in figure 2 b, is based on a section of the road map of Manhattan, New York, and is a more complex network containing several interconnected intersections. The specific road map used in our experiments is a 700 m^2 section centered around the corner of Waring and Woodhull Avenues. The map has been imported using SUMO's web-wizard. This scenario means to evaluate the effect of efficient exploration in a more complex traffic profile, where the agent may observe and consider the behavior of neighboring intersections. Manhattan enjoys a grid road-map design, that for the purpose of this work serves as both realistic and practical to use.

The red squares marked in figure 2 point to the two intersections given to the agent to control, as two separate experiments sets in this scenario. These intersections were chosen due to their encapsulation of different traffic profiles. The top intersection, Waring-Woodhull, presents a gentler form of traffic with significantly lower vehicle speed average. The bottom intersection, Eastchester-Waring, experiences much higher average vehicle speed, and more strongly resembles a central road. The throughput of both intersections is rather similar, with a slightly heavier load going through Eastchester-Waring. In every experiment, all intersections in the network except the one controlled by the agent are controlled by SUMO. To generate traffic for the Manhattan scenario, a random routes generator is used, based on real population distributions in the area, for the time of day 08:30 AM to 09:00 AM. The traffic data is imported using SUMO's web-wizard as well.

Scenario 3: Simulation of custom traffic in Manhattan, New York A third scenario is introduced in order to evaluate exploration under heavier traffic settings. This scenario uses the same map, and experiments with the same intersections illustrated in figure 2 b. However, in this scenario a random traffic generator is used for traffic generation, in order to introduce much heavier traffic loads than the ones generated to simulate real traffic. Again, two different sets of experiments are done in this scenario, one on each of the two marked intersections in figure 2 b. The difference of the traffic profiles between the two intersections is similar to the one in the second scenario: the top intersection enjoys slower average speed, and the bottom faster.



(a) The basic grid map.

(b) The Manhattan map. The top red box marks the intersection Waring-Woodhull, and the bottom the intersection Eastchester-Waring.

Fig. 2: The road maps used in the traffic scenarios.

Observation space specification An important difference between the scenarios is the observation space provided to the agent in each one. The observation space provided in the experiments done with the grid scenario and the Eastchester-Waring intersection in either scenario, was $50\,\mathrm{m}^2$. This is done because in all three a larger observation space would be outside the bounds of the environment. Due to the structure of the scenario, it was possible to provide the agent with a larger observation space in the experiments done with the Waring-Woodhull intersection in either scenario. An observation space of $84\,\mathrm{m}^2$ was chosen, to balance complexity (the larger the observation, the more complex the learning) with observation distance. This is done to allow the agent access to more information, which (1) contains the adjacent intersections, enabling the agent to take their behavior into account, and (2) providing the agent with the ability to react to incoming traffic earlier, as a result of the larger observation space.

5 Results

This section presents and analyzes the results obtained in the experiments described in section 4, divided between the different scenarios investigated and intersections controlled.

For each set of experiments, the results presented are the episodic rewards attained in the evaluations, as described in section 4.1. This is presented alongside a plot of the 95% confidence interval of the mean, computed using the standard error of the mean (SEM) [2] of the different experiments for two sample agents. These agents are chosen independently for each experiment: the one that performed, on average, the best, and the one that performed the worst. This is done to illustrate how significant are the differences observed between the evaluations of the different agents in each experiment. Only two agents are presented in order to reduce clatter in the plot. A simple moving average (SMA) of window size 5 is applied to the data presented in order to smoothen the stochastic effect, to facilitate visual analysis of the results.

5.1 Scenario 1: The grid

The results of evaluating the performance of six different agents against the grid scenario can be found in figure 3. The agents' policies are evaluated every five learning episodes, and averaged over three evaluation episodes. Additionally, the 95% confidence intervals of the means of the two sample agents are presented.

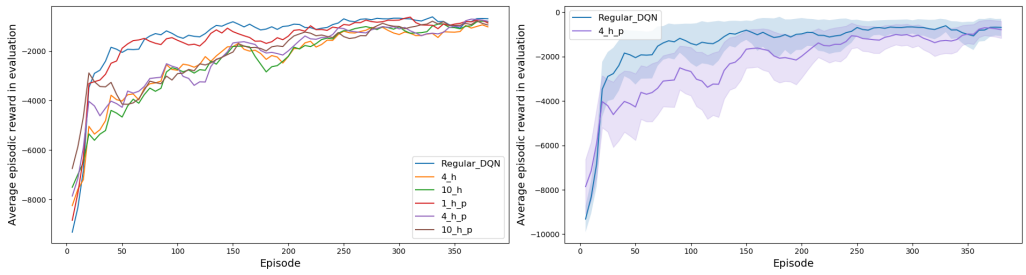


Fig. 3: Evaluating the agents against the grid scenario. The left figure presents the evaluations. The number describes the size of the bootstrap sample (number of heads), and the p whether a randomized prior has been incorporated. The right figure presents the same for two chosen agents, including the 95% confidence interval of the mean.

Figure 3 illustrates that while all agents learn, the agents that appear to have the sharpest learning rates are the regular DQN with and without prior function applied. However, as can be seen in the right plot in figure 3, there is some overlap in the confidence intervals of their means.

5.2 Scenario 2: Simulation of real traffic in Manhattan, New York

The results of evaluating the agents against the Manhattan scenario simulating real traffic can be found below, separately for each set of experiments, controlling each of the two intersections marked in figure 2 b. The agents' policies are evaluated every training episode, over two evaluation episodes, and averaged.

Traffic of low average speed Figure 4 presents the results of experimenting control of the intersection Waring-Woodhull, the top of the two intersections in figure 2 b, capturing a traffic profile of slower average speed.

As observable, the difference between the evaluation scores in relation to the confidence interval of the means, is mostly negligible, with the exception that the regular DQN achieves inferior scores prior to episode 20. However, these scores are still well within the confidence interval of the means of the other agent's, and thus cannot be considered significant. This behavior is attributed to the simplicity of the traffic profile, in relation to the amount of information accessible to the agent in this scenario. As mentioned in section 4.2, the observation space of the agents in this experiment is $84\,\mathrm{m}^2$. While the scenario can be viewed as complex (several interconnected intersections, whose behaviors directly influence each other's traffic), which can translate to the learning process being more difficulty, the volume of the traffic, including the average speed, is rather low, and thus the policy required is not complex.

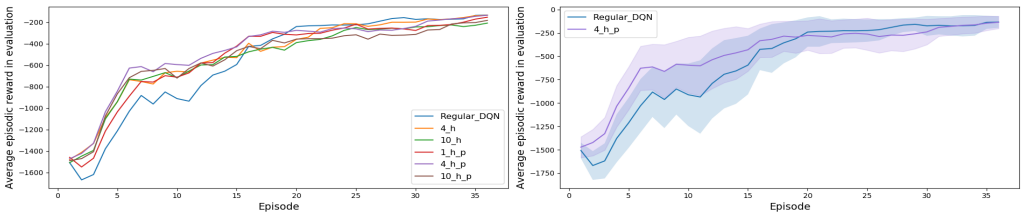


Fig. 4: Evaluating the different agents against the Waring-Woodhull intersection, simulating real traffic. The left figure presents the evaluation of the different agents. The number describes the size of the bootstrap sample (number of heads), and the p whether a randomized prior has been incorporated. The right figure presents the same results for a sample of the agents, including the 95% confidence interval of the mean.

Traffic of high average speed Figure 5 presents the results of experiments done controlling the second intersection, Eastchester-Waring. Eastchester-Waring enjoys both traffic of higher average speed, as well as higher traffic loads. No significant difference in the performance of the different agents is observed. This is further illustrated with the confidence interval of the means presented in figure 5 and their overlap.

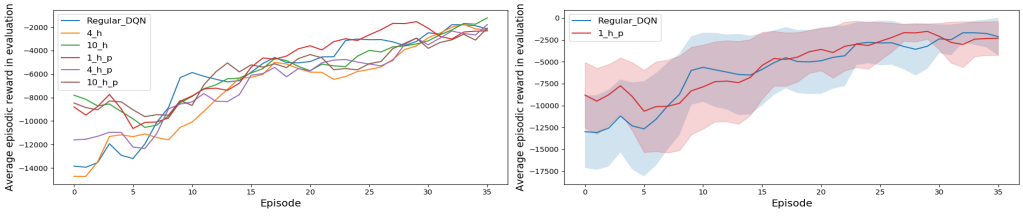


Fig. 5: Evaluating the different agents against the Eastchester-Waring intersection, simulating real traffic. The left figure presents the evaluation of the different agents. The number describes the size of the bootstrap sample (number of heads), and the p whether a randomized prior has been incorporated. The right figure presents the same results for a sample of the agents, including the 95% confidence interval of the mean.

5.3 Scenario 3: Simulation of custom traffic in Manhattan, New York

The results of evaluating the agents against the Manhattan scenario simulating random, heavy traffic can be found below. The agents' policies are evaluated every second training episodes, and averaged over two evaluation episodes.

Traffic of low average speed The results for the intersection Waring Woodhull are presented in figure 6. While the differences in the learning between most of the different agents appears negligible, all of them appear to outperform the regular DQN agent. However, as illustrated by the right plot in figure 6, the confidence intervals still have some overlap.

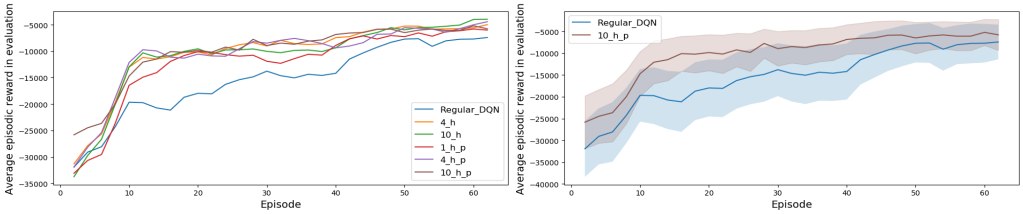


Fig. 6: Evaluating the different agents against the Waring-Woodhull intersection in the custom scenario. The left figure presents the evaluation of the different agents. The number describes the size of the bootstrap sample (number of heads), and the p whether a randomized prior has been incorporated. The right figure presents the same results for a sample of the agents, including the 95% confidence interval of the mean.

Traffic of high average speed The results of experimenting control over the intersection Eastchester Waring are presented in figure 7. No significant difference is observed between the evaluations of the different agents. This is illustrated strongly by the confidence intervals, and their overlap, presented in the right plot in figure 7.

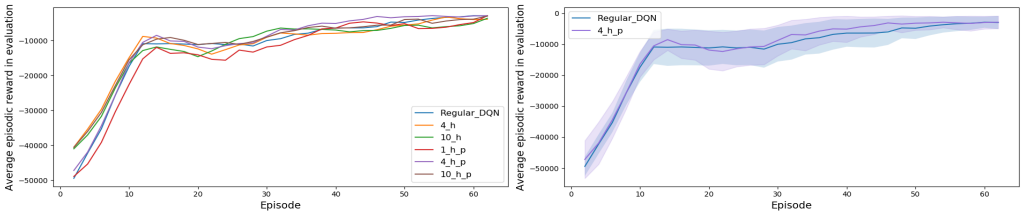


Fig. 7: Evaluating the different agents against the Eastchester-Waring intersection in the custom scenario. The left figure presents the evaluation of the different agents. The number describes the size of the bootstrap sample (number of heads), and the p whether a randomized prior has been incorporated. The right figure presents the same results for a sample of the agents, including the 95% confidence interval of the mean.

5.4 Analysis

Many different factors can influence the results obtained by the different agents, ranging from the reward function, the size of the observation space and its resolution, the state abstraction, the complexity of the traffic scenario and sheer stochasticity. The type and complexity of the traffic scenario, along with the observation space, are the main parameters that will be considered in this analysis.

Here, the complexity of the scenario, while not defined exactly, considers the volume and variety in the traffic and the number of surrounding interconnected intersections. In particular, the "grid" scenario can be viewed as having lower complexity, as it has a medium traffic volume, of low speeds and no neighbouring intersections. In the experiments done in the scenario simulating real traffic, both intersections could be viewed as having higher complexity: "Waring Woodhull", which is surrounded by a number of neighbouring intersections, with lower traffic volume and a slower speed; and Eastchester-Waring, which is surrounded by a similar number of neighbouring intersections, with somewhat higher traffic volume with significantly higher average speed. Last, the scenario simulating custom traffic could be viewed as having higher complexity as well, as it has similar parameters to the simulation of real traffic, with the exception of significantly increased traffic loads. In addition, it is worth mentioning that in the experiments done with the Waring Woodhull intersection the observation space was larger than in any of the other experiments, as specified in section 4.2.

It appears that when the complexity of the traffic is low along with a smaller observation space (the grid scenario), the simpler agents converge much faster, and more stably, to a highly evaluated policy. However, when the agent is given sufficient information of sufficient complexity (the Waring Woodhull intersection in both scenarios), deep exploration approaches are able to outperform the ϵ-greedy approach, by achieving faster learning. This may imply that advanced exploration approaches may play a significantly more critical role in much more sophisticated scenarios, such as an intersection balancing many lanes, experiencing many different traffic profiles, and provided a large observation space. The differences observed were generally small and within a 95% confidence interval of the means however, and as such, the strength of this implication is limited.

We note, though, that for the Eastchester-Waring intersection no improvements due to more sophisticated exploration were observed, even though the traffic patterns are more complex here than in the simple grid. The reason for this could be that as a result of the faster traffic and the smaller observation space, along with sufficient complexity in the road map and variety in the loads and directions of the traffic, the different costs and gains from the different exploration approaches balanced out.

6 Discussion

The results obtained suggest a link between the complexity of the scenario, the information accessible to the agent and the gain from deep exploration, as discussed in section 5.4. However, the advanced exploration approaches investigated are expected to have significant gain especially when utilized in environments with scarcity of significant positive rewards, and abundance of smaller negative rewards [11], [12]. As a consequence, the reward function used can almost directly dictate the gain from different exploration strategies. The reward function used was not designed for any specific exploration approach. It is therefore plausible that under a tailored reward function the gain from deep exploration would have been much

higher. This can be especially relevant in the case of reward functions that are known for promoting good policies, while hindering learning.

7 Conclusions and Future Work

This paper investigated the value of deep exploration in the setting of traffic light control, by comparing agents using different exploration approaches in three different traffic scenarios of rising complexity. Specifically, the state of the art approaches Bootstrapped DQN [12] and randomized prior functions [11] were compared to a baseline ϵ-greedy approach. This, to facilitate better deep RL in traffic light control, by identifying the value of evolved exploration approaches in this setting, such as higher sample efficiency or higher final policy score.

The results presented in this paper suggest a link between the complexity of the traffic scenario, the size of the observation space of the agent, and the gain from efficient exploration, achieved with Bootstrapped DQN and randomized prior functions, under a specific parameters configuration. Specifically, the more complex the scenario and the larger the observation space, the more significant the gain observed from efficient exploration.

The results presented leave the following open questions, however. What is the exact relation between the gain from efficient exploration and the complexity of the scenario? How sensitive is this relation to specific parameter configurations, and specifically under parameters optimized for the specific setting? Does this conclusion apply for other exploration approaches? These questions are left for future work.

Acknowledgements

We would like to thank the students in the research-project group, Pepijn Tersmette, Cian Jansen, Emanuel Kuhn & Chris van der Werf, for their ready assistance and availability for discussion, along with their contributions to the creation of the traffic scenarios and the implementation of the reward function used. Additionally, we would like to thank the TUDelft ILDM research group for their helpful comments and support. An additional thanks goes to the TUDelft for granting us access to the INSY cluster, which was a crucial computational resource used in this research.

This project had received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

References

1. Bakker, B., Whiteson, S., Kester, L., Groen, F.C.: Traffic light control by multiagent reinforcement learning systems. In: Interactive Collaborative Information Systems, pp. 475–510. Springer (2010)
2. Barde, M.P., Barde, P.J.: What to use to express the variability of data: Standard deviation or standard error of mean? Perspectives in clinical research **3**(3), 113 (2012)

3. Ciosek, K., Vuong, Q., Loftin, R., Hofmann, K.: Better exploration with optimistic actor critic. In: Advances in Neural Information Processing Systems. pp. 1785–1796 (2019)
4. Coşkun, M., Baggag, A., Chawla, S.: Deep reinforcement learning for traffic light optimization. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 564–571. IEEE (2018)
5. Efron, B., Tibshirani, R.J.: An introduction to the bootstrap. CRC press (1994)
6. Fortunato, M., Azar, M.G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al.: Noisy networks for exploration. arXiv preprint arXiv:1706.10295 (2017)
7. François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., Pineau, J.: An introduction to deep reinforcement learning. arXiv preprint arXiv:1811.12560 (2018)
8. Hansen, J.: bootstrap_dqn. GitHub repository (2019)
9. Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., WieBner, E.: Microscopic traffic simulation using sumo. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 2575–2582. IEEE (2018)
10. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
11. Osband, I., Aslanides, J., Cassirer, A.: Randomized prior functions for deep reinforcement learning. In: Advances in Neural Information Processing Systems. pp. 8617–8629 (2018)
12. Osband, I., Blundell, C., Pritzel, A., Van Roy, B.: Deep exploration via bootstrapped dqn. In: Advances in neural information processing systems. pp. 4026–4034 (2016)
13. Van der Pol, E., Oliehoek, F.A.: Coordinated deep reinforcement learners for traffic light control. Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016) (2016)
14. Rezzai, M., Dachry, W., Mouataouakkil, F., Medromi, H.: Reinforcement learning for traffic control system: Study of exploration methods using q-learning. International Research Journal of Engineering and Technology **4**(10), 1838–1848 (2017)
15. TO, H.R., Barker, M.M.: White paper european transport policy for 2010: time to decide (2001)
16. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Thirtieth AAAI conference on artificial intelligence (2016)
17. Watkins, C.J., Dayan, P.: Q-learning. Machine learning **8**(3-4), 279–292 (1992)
18. Wei, H., Zheng, G., Yao, H., Li, Z.: Intellilight: A reinforcement learning approach for intelligent traffic light control. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2496–2505 (2018)

Does the dataset meet your expectations? Explaining sample representation in image data*

Dhasarathy Parthasarathy[1,2] and Anton Johansson[2]

[1] Volvo Group, Sweden
dhasarathy.parthasarathy@volvo.com
[2] Chalmers University of Technology, Sweden
johaant@chalmers.se

Abstract. Since the behavior of a neural network model is adversely affected by a lack of diversity in training data, we present a method that identifies and explains such deficiencies. When a dataset is labeled, we note that annotations alone are capable of providing a human interpretable summary of sample diversity. This allows explaining any lack of diversity as the mismatch found when comparing the *actual* distribution of annotations in the dataset with an *expected* distribution of annotations, specified manually to capture essential label diversity. While, in many practical cases, labeling (samples → annotations) is expensive, its inverse, simulation (annotations → samples) can be cheaper. By mapping the expected distribution of annotations into test samples using parametric simulation, we present a method that explains sample representation using the mismatch in diversity between simulated and collected data. We then apply the method to examine a dataset of geometric shapes to qualitatively and quantitatively explain sample representation in terms of comprehensible aspects such as size, position, and pixel brightness.

Keywords: Sample selection bias · Explainability · Outlier detection.

1 Introduction

Choosing the right data has always been an important precondition to deep learning. However, with increasing application of trained models in systems which are required to be dependable ([20], [2]), there is increasing emphasis on making this choice well-informed ([4], [36]). Consider the perception system of a self-driving vehicle which is partially realized using deep learning and is expected to dependably detect pedestrians. To ensure that the system meets such an expectation, it is necessary to choose training and validation sets that adequately cover critical scenarios ([31], [34]) like residential areas and school zones, where the vehicle is likely to meet pedestrians. Choosing, conversely, a dataset that contains only scenes of motorway traffic, which does not cover many scenarios involving pedestrians, is likely to produce a trained model that violates

* Work supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

expectations on pedestrian detection. Scenarios covered by a dataset may be considered sufficient when samples of adequate variety are represented in it. With practical image datasets typically being high-dimensional and large, posing and evaluating explicit conditions on the adequacy of sample representation is not straightforward.

Interpretable assessment of sample representation Consider a traffic dataset $\mathcal{S}$ of images $X_i \sim P(X|Y)$ and annotations $Y_i \sim P(Y)$. A major practical concern in such datasets is whether it adequately represents corner cases like intersections with stop signs, roundabouts with five exits, etc. With the true/target distribution of traffic scenes $P(X, Y)$ clearly containing instances of such cases, *any under-representation* in $\mathcal{S}$ can be broadly framed as shortcomings in data collection and processing, otherwise known as *sample selection bias*([37]). Given that the dataset is eventually used to train a model that is deployed in a safety-critical system, engineers may actively seek to properly comprehend and account for such bias. But how does one express such bias in human interpretable terms? One clue comes from annotations $Y_i \sim P(Y)$. In typical traffic datasets, Y encodes object class labels and bounding box positions. If necessary and feasible, Y can be expanded to contain information such as location, lighting conditions, weather conditions, etc. When Y is adequately detailed, the distribution of annotations $P_{\mathcal{S}}(Y)$ clearly becomes a reasonable, low-dimensional, and therefore a human interpretable measure of sample representation in $\mathcal{S}$. Engineers can exploit this notion to *specify* a distribution of annotations $P_{\mathcal{T}}(Y)$, expressing the sample representation that is *expected* in the dataset. While the target distribution of annotations $P(Y)$ may be unknowable, $P_{\mathcal{T}}(Y)$ is an explicit declaration of the sub-space that the dataset is expected to cover at the minimum. If $\mathcal{S}$ is equivalently labeled, then selection bias (and thereby sample under-representation) is simply given by the mismatch between expectations $P_{\mathcal{T}}$ and reality $P_{\mathcal{S}}$. In practice, however, due to the effort and expense involved in labeling, $\mathcal{S}$ may either lack labels or may be completely unlabeled, meaning that $P_{\mathcal{S}}(Y)$ is often unavailable. Combining simulation, outlier detection, and input attribution, we show that it is possible to explain sample representation in a comprehensible low-dimensional form, even when annotations are not explicitly available in $\mathcal{S}$.

Contributions Delving into the less-explored area of *explaining* sample representation in a dataset, we demonstrate a method that

- explains sample representation in interpretable terms for annotated data
- uses parametric simulation and outlier detection to do the same for non-annotated data

In addition to visualization, we propose a quantitative explanation of sample under-representation using an *overlap index*. Also, unlike existing methods that mainly address imbalances in available data, ours can explain gaps in the availability of data. Such an explanation helps engineers better understand data as a crucial ingredient of the training process. Downstream, this helps them re-asses data collection methods and to verify, reason, or argue about – at times a re-

quirement for standards compliance [3] – the overall dependability of the model trained with this data. Data and code used in this work are publicly available[3].

2 Explaining sample representation using annotations

Visualizing sample representation We now introduce a simple running example of examining sample representation in a dataset $\mathcal{S}$ containing images of two hand-drawn shapes[4] – circles and squares (Figure 1). With the shape as the sole available label, one can define $\mathcal{S} = \{(X_i, Y_i^1)\},\ i = 1...N$, where X_i is a grayscale image of size $(128, 128)$ and $Y_i^1 \in K = \{0, 1\}$ is the shape label, corresponding to circle and square respectively. Understanding sample representation in this dataset may be necessary when it is a candidate for training a model that, for example, either recognizes or generates shapes. To ensure dependable model performance, system designers may want to confirm that images of adequate variety are represented in $\mathcal{S}$. In a dataset of grayscale geometric shapes, it is intuitive to analyze sample representation in terms of concerns such as the size and position of the shapes on the image canvas, and the average brightness of pixels in the shape. All these concerns can be captured by defining a 6-d annotation vector $Y = (Y^1, ..., Y^6)$, including shape-type, which is known. With $\mathcal{U}$ denoting the discrete uniform distribution, designers can begin with defining an expected spread of shape-size using a latent label $Y^S \sim \mathcal{U}\{30, 120\}$, denoting the side-length in pixels of a square box bounding the shape. This can be followed by defining expectations on the spread of (i) the top-left corner of the bounding box, $Y^2, Y^3 \sim \mathcal{U}\{0, 128 - Y^S\}$, (ii) the bottom-right corner of the bounding box $Y^4, Y^5 \sim \mathcal{U}\{Y^S, 128\}$, and (iii) the average pixel brightness $Y^6 \sim \mathcal{U}\{100, 255\}$. Put simply, $P_{\mathcal{T}}(Y)$ expects shapes of a specified range of sizes and brightness to be uniformly represented in the dataset $\mathcal{S}$. All positions are also expected to be uniformly represented, as long as the shape can be fully fit in the image canvas.

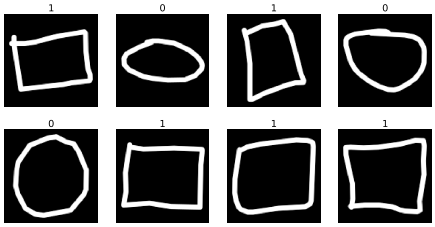


Fig. 1: Samples from the dataset $\mathcal{S}$. Only the class label Y^1 is available

To illustrate the idea of explaining sample representation using annotations, an automatic labeling scheme $Y_i = L(X_i)$ is used to produce complete 6-d annotations for X_i. For circles and squares, it is easy to define a scheme that looks

[3] https://github.com/dhas/SpecCheck
[4] Collected from Quick, Draw! with Google – https://quickdraw.withgoogle.com/data

at the extent of the shape and draws bounding boxes. The average brightness is given by the mean of non-zero pixels in the canvas. The availability of labels Y_i helps assemble the actual distribution of samples in the dataset $P_{\mathcal{S}}(Y)$, allowing direct comparison with expectations $P_{\mathcal{T}}(Y)$. Jointly visualizing label distributions for each shape (Figure 2) shows that, along all design concerns Y^j, the spread of $P_{\mathcal{T}}$ (marked black) is much wider than the very narrow $P_{\mathcal{S}}$ (marked red). This shows that, while $P_{\mathcal{T}}$ expects shapes of a broad range of sizes, positions and brightness to be represented, $P_{\mathcal{S}}$ is clearly biased and massively over-represents large and bright shapes located in the center on the canvas. As long as the annotation vector Y is of manageable length, joint visualization becomes an interpretable qualitative explanation of sample representation in the dataset.

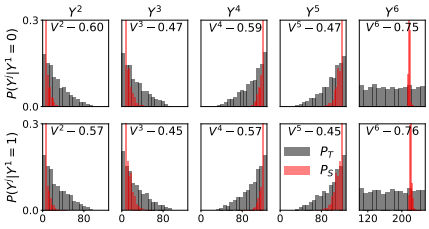


Fig. 2: Explaining sample representation

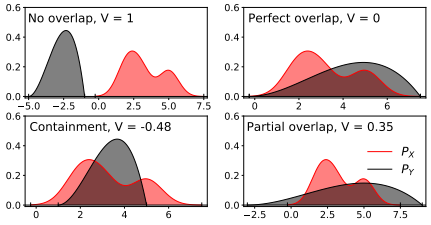


Fig. 3: Illustration of $V(P_X, P_Y)$

Quantifying sample representation By framing sample selection bias, and thereby sample under-representation, as the mismatch between expected and true label probability distributions, it becomes possible to quantify it using measures of statistical similarity. Choosing the right measure, however, requires a proper understanding of the nature of each distribution. Having calculated it using true labels of each sample, it is clear that $P_{\mathcal{S}}(Y)$ represents the actual sample distribution in $\mathcal{S}$. The distribution of expectations $P_{\mathcal{T}}$ is of a slightly different nature and, to better understand it, let us consider the expectation $P_{\mathcal{T}}(Y^6) = \mathcal{U}\{100, 255\}$, placed on the representation of average brightness of shapes in the dataset. While the expectation on brightness being spread between specified lower and upper limits is strict, imposing the spread to be uniform is arbitrary. This is a deliberate measure of simplification to ease the considerable burden in modeling expectations $P_{\mathcal{T}}$ and let it capture the critical range of interest in the target distribution. Put simply, expected sample representation is primarily encoded by the *support* (1) of $P_{\mathcal{T}}$. By specifying strict support, but arbitrary distribution of mass, sample representation can be quantified as the level of *overlap* between the actual sample distribution $P_{\mathcal{S}}$ and the expected sample representation $P_{\mathcal{T}}$. To achieve this, we propose an overlap index $V(P_X, P_Y)$ (2), which is a measure of whether the supports of two distributions are similar. With set difference Δ and 1-d Lebesgue measure (length) of a set λ, V is essentially the Steinhaus distance [11] with an added term I to make $-1 < V < 0$ indicate containment of P_Y within P_X. When not contained, for some positive likelihood

in both distributions, as illustrated in Figure 3, $V = 0$ when they exactly overlap, $V = 1$ when they do not overlap, and $0 < V < 1$ when the overlap is partial. Indices $V^j(P_{\mathcal{T}})$ (3) quantitatively measure the level of overlap between true and expected distributions for each label. Complementing the visual explanation, overlap indices $0.4 < V^j(P_{\mathcal{T}}) < 1$ seen in Figure 2, indicate that there is only slight partial overlap between expectations and reality, confirming notable sample selection bias and, therefore, significant sample under-representation.

$$R_X = \{x \in \mathbb{R} : P_X(x) > 0\} \tag{1}$$

$$V(P_X, P_Y) = I\ \frac{\lambda(R_X\ \Delta\ R_Y)}{\lambda(R_X\ \cup\ R_Y)},\quad I = \begin{cases} -1 & R_Y \subset R_X \\ +1 & otherwise \end{cases} \tag{2}$$

$$V^j(P) = V(P_{\mathcal{S}}(Y^j\ |Y^1), P(Y^j\ |Y^1)),\quad j = 2...6 \tag{3}$$

It is therefore clear that, given the expected representation and actual distribution of labels in the dataset, it is possible to comprehensibly explain sample under-representation both visually and quantitatively. However, the overlap index, which eschews mass and uses only support, is an incomplete measure of sample selection bias, the pros and cons of which is discussed in Section 4.

3 Explaining sample representation using simulation

The dataset $\mathcal{S}$ contains information X_i in the image domain, while lacking information Y_i in the annotations domain. Expectations, on the contrary, are expressed using annotations $\hat{Y}_i \sim P_{\mathcal{T}}(Y)$, but lacks images. It is this gap in information that prevents estimation of sample under-representation by direct comparison. There are two possible ways to bridge this gap, one of which is the labeling scheme $Y_i = L(X_i)$ introduced earlier. Another way could be to generate images $\hat{X}_i = G(\hat{Y}_i)$, which is essentially *parametric simulation*. In this case of circles and squares, it is possible to use a graphics package[5] to draw shapes using size, position, and brightness labels as parameters. We, in fact, choose this simple dataset because both labeling and simulation of samples are easy, helping illustrate both ways of bridging the gap and cross-checking the plausibility of estimating sample representation. In many practical cases, however, the right method to bridge the gap is difficult to judge since the relative expense is domain and problem specific. Addressing those numerous instances where unlabeled data is available and labeling is expensive, we now show that it is possible to bridge the gap using simulation. This is done using a two-step process, described below, of (i) detecting outlier annotations and (ii) estimating marginal sample representation.

Step 1 - Detecting outlier annotations To a dataset that mainly contains large, centered shapes, can simulated small off-centered shapes appear as outliers? In order to explore this simple notion, we pose the following outlier hypothesis - *a test annotation $\hat{Y}_i$, that is unlikely to be observed in $\mathcal{S}$, maps to a simulated test sample $\hat{X}_i = G(\hat{Y}_i)$, that appears as an outlier to $\mathcal{S}$.* Bridging the gap

[5] We use OpenCV – https://opencv.org/

by simulating shape images that follow specified expectations $P_{\mathcal{T}}$, the problem of detecting sample selection bias turns into one of detecting outlier images. The hypothesis is realized by an outlier detector $E_{\mathcal{S}}$ (Figure 4) that samples test annotations from $P_{\mathcal{T}}$ and maps them into images using a simulator, creating a test set $\mathcal{T} = \{(\hat{X}_i, \hat{Y}_i)\},\ i = 1...M$ (examples in Figure 5). Following [17], the subsequent assessment of whether under-represented simulated images appear as outliers to $\mathcal{S}$ is done using the predictive certainty of a shape label classifier $F(X) = P_{\mathcal{S}}(Y^1|X;\theta)$, trained on the dataset $\mathcal{S}$. The complete detector of outlier annotations $E_{\mathcal{S}}$ is formally described below in (4), where F_k is the logit score for the k^{th} shape and T is the temperature parameter which, as shown later, eases the detection process. With F using a softmax output layer, we use maximum softmax score as the measure of certainty. Put simply, with sets of outlier and familiar annotations (5), the outlier hypothesis asserts that a good detector $E_{\mathcal{S}}$ assigns low scores S_i for outlier annotations $\hat{Y}^-$ and high scores for familiar ones $\hat{Y}^+$.

$$S_i = E_{\mathcal{S}}(\hat{Y}_i, F, T) = \max_{k \in K} \frac{exp(F_k(G(\hat{Y}_i))/T)}{\sum_{k \in K} exp(F_k(G(\hat{Y}_i))/T)}, \hat{Y}_i \sim P_{\mathcal{T}}(Y), K = \{0, 1\} \quad (4)$$

$$\hat{Y}^- = \{\hat{Y}_i : P_{\mathcal{S}}(\hat{Y}_i) = 0\}, \qquad \hat{Y}^+ = \{\hat{Y}_i : P_{\mathcal{S}}(\hat{Y}_i) > 0\} \quad (5)$$

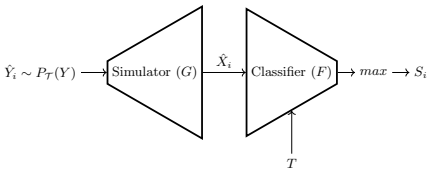


Fig. 4: Detecting outlier annotations

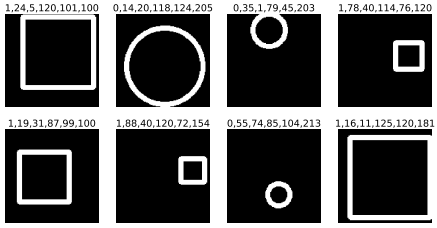


Fig. 5: Samples $\hat{X}_i$ from the test set $\mathcal{T}$

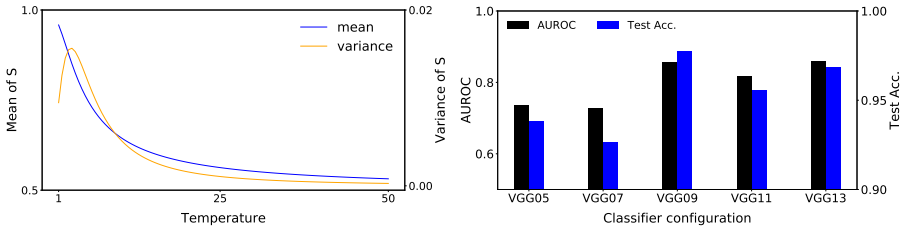
To test the outlier hypothesis, four variants of the classifier F, all of which follow the VGG architecture [33], are used. Classifiers mainly differ in the number of layers, with VGG05 (5 layers) and VGG13 (13 layers) being the shallowest and deepest respectively. Each F is trained[6] for 5 epochs on $\mathcal{S}$ with 50k samples using the Adam optimizer [18] to achieve validation accuracy (on a separate set of 10k samples) greater than 97%. However, [14] shows that deep neural nets tend to predict with high confidence, making raw maximum softmax scores poor measures of predictive certainty, and a simple way to mitigate this is temperature scaling, i.e. setting $T > 1$, in (4). As seen in Figure 6a, scores S_i are tightly clustered at $T = 1$ with relatively low variance, which makes it difficult to identify differences in predictive certainty between familiar and outlier annotations. There is, however, a range of temperatures at which scores are better spread and can exaggerate these differences. While a temperature that maximizes the

[6] Each classifier trains within 10 – 15 minutes on an NVidia GTX 1080 Ti GPU

variance of the score distribution seems appropriate, as seen in Figure 6a, scaling also reduces its mean. Therefore a safeguard may be necessary to prevent the mean certainty score from reducing to a level that questions the confidence of predictions. These twin requirements can be achieved by the search objective (6), which ensures a good spread in scores S_i, while keeping its mean close to the chosen safeguard S^T.

$$\begin{aligned} T^* &= \underset{T}{argmin}\ L^T - L^V, \quad L^T = \left(\mu_S - S^T\right)^2 \\ L^V &= \frac{\sum_{i=1}^{M}\left(E_{\mathcal{S}}(\hat{Y}_i, F, T) - \mu_S\right)^2}{M}, \quad \mu_S = \frac{\sum_{i=1}^{M} E_{\mathcal{S}}(\hat{Y}_i, F, T)}{M} \end{aligned} \tag{6}$$

Upon temperature scaling with T^*, the effectiveness of the detector $E_{\mathcal{S}}$ in separating outlier annotations $\hat{Y}^-$ from familiar ones $\hat{Y}^+$ can be measured using the Area Under Receiver Operating Characteristic (AUROC). This is shown for each F, averaged over 5 separate training runs, in Figure 6b. Based on an informal grading scheme for classifiers using AUROC score suggested in [17][7], detectors using VGG05 and VGG07 receive a 'fair' grade in identifying outlier annotations, while the deeper networks get 'good' grades. The best outlier detectors, with AUROC ≈ 0.85, are those with F as VGG09 and VGG13. These results clearly endorse the viability of the outlier hypothesis that simulated images that are under-represented in $\mathcal{S}$, in terms of specified design concerns, appear as outliers to the right classifier trained on $\mathcal{S}$. While $P_{\mathcal{S}}$, derived from labeling, is used as a benchmark to test the outlier hypothesis, it is important to observe that (i) classifiers that are good at outlier detection are, as seen in Figure 6b, those that have the highest accuracy in predicting shape labels on the test set $\mathcal{T}$, and (ii) the temperature T^*, at which the classifiers become good outlier detectors, depends only upon the statistical properties of scores S_i. Together, these observations mean that a good detector of under-represented annotations can be assembled using only simulation, without any need for labeling.



(a) Effect of temperature scaling on the distribution of uncertainty scores F=VGG13

(b) AUROC for detecting outlier annotations per classifier at $T = T^*$ and $S^T = 0.7$

Fig. 6: Testing the novelty hypothesis

[7] Quality of classification based on AUROC score - 0.9—1: Excellent, 0.8—0.9: Good, 0.7—0.8: Fair, 0.6—0.7: Poor, 0.5—0.6: Fail

Step 2 - Estimating marginal sample representation As presented in Section 2, we seek to comprehensibly explain sample representation in the dataset $\mathcal{S}$ of geometric shapes on the basis of intuitive design concerns like size, position, and brightness. However, the detector $E_{\mathcal{S}}$ can only assess whether a single combined 6-d test annotation is an outlier. To assess, for example, the diversity of shape sizes in the dataset, independent of position, we turn to techniques of input attribution. Given the detector $E_{\mathcal{S}}$, attribution techniques estimate the contribution of each input label $\hat{Y}_i^j$ to its outlier score S_i. Among proposed methods for input attribution [29], one promising framework is Shapley Additive Explanations (SHAP)[25]. Using principles of cooperative game theory, SHAP estimates *marginal influence* ϕ_i^j (7), which indicates how label $\hat{Y}_i^j$ independently influences the uncertainty score S_i.

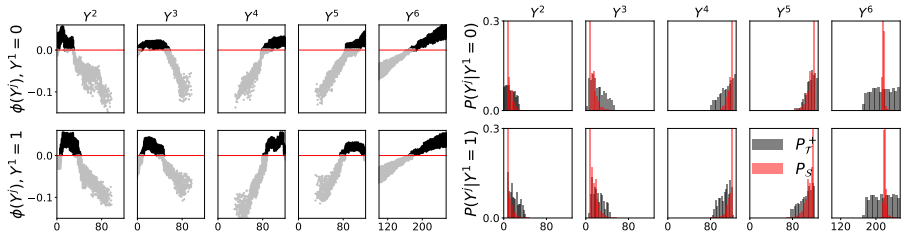
$$S_i = E_{\mathcal{S}}(\hat{Y}_i, F, T) = \phi^0 + \sum_{j=2}^{6} \phi_i^j \tag{7}$$

In satisfying an additive property, SHAP values are also semantically intuitive, with negative, positive, and zero values of ϕ_i^j respectively indicating negative, positive, and neutral influence of label $\hat{Y}_i^j$ on the score S_i. The outlier hypothesis verified earlier implies that outlier (familiar) annotations tend to have a lower (higher) certainty score S_i. Therefore SHAP value $\phi_i^j > 0$, which indicates that the individual label value $\hat{Y}_i^j$ tends to improve S_i, becomes an indicator of that label being represented in $\mathcal{S}$. Through a campaign directed by the test set $\mathcal{T}$, which systematically covers the specified range of scenarios $P_{\mathcal{T}}$, non-negative SHAP values identify sample representation in the dataset $\mathcal{S}$ in terms of each individual label. This can be seen in Figure 7a, where label values with a high incidence of non-negative SHAP values (marked black) are likely to be represented in $\mathcal{S}$. This directly allows estimating the likelihood of test label $Y^j = l,\ Y^j \sim P_{\mathcal{T}}$ being represented in the set $\mathcal{S}$ as the proportion of test labels $\hat{Y}_i^j$, in a sufficiently small interval δ around l, whose SHAP values are non-negative.

$$\begin{gathered} P_{\mathcal{T}}^{+}(Y^j = l \mid Y^1 = k) = \frac{|\{\hat{Y}_i^j : \phi_i^j \geq 0,\ \hat{Y}_i^j \in Y^l\}|}{|\{\hat{Y}_i^j : \phi_i^j \geq 0\}|},\ j = 2...6,\ \hat{Y}_i \in \hat{Y} \\ Y^l = \{l - \delta, l + \delta\}, \qquad \hat{Y} = \{\hat{Y}_i : \hat{Y}_i^1 = k\},\ k \in K \end{gathered} \tag{8}$$

Assessing the explanation By expressing expected diversity $P_{\mathcal{T}}$ in terms of specified design concerns, the two-step process, using a simulated test set, identifies sample representation in each concern using non-negative influence on predictive certainty. From the original broadly spread expectations $P_{\mathcal{T}}$ (Figure 2), the process correctly eliminates a significant amount of outliers in each label dimension, producing $P_{\mathcal{T}}^{+}$ (Figure 7b). $P_{\mathcal{T}}^{+}$ shows label values likely to be observed in the dataset $\mathcal{S}$ and has a roughly similar spread as the actual distribution $P_{\mathcal{S}}$. Also, using a test set with M=10k samples, the process estimates sample representation in a much larger dataset with N=50k samples.

Introduced originally in Section 2 to quantify bias between expected and actual distributions of annotations, the overlap index V is also suitable for measuring similarity between $P^+_{\mathcal{T}}$ and $P_{\mathcal{S}}$. This helps quantify the effectiveness of estimating sample representation using simulation. The visual observation that P^+_T is a better estimate of true sample distribution, compared to the broad range of expectations $P_{\mathcal{T}}$, is confirmed by better a mean overlap score $V^j(P^+_{\mathcal{T}})$ (see Table 1), over all labels and shapes, compared to mean $V^j(P_{\mathcal{T}})$. While this holds true for both classifier instances shown in the table, the detector using F=VGG13 at $T = T^*$, which has the best AUROC score in detecting outliers, produces the closest estimate with a mean overlap score of 0.27. VGG05, with poorer AUROC, has a weaker average overlap score of 0.39. The close correlation between AUROC and V further confirms the plausibility of estimating marginal sample representation using SHAP scores. This shows that, while facing an expensive labeling process, with the right means of parametric simulation, one can conduct a campaign from a low-dimensional space of specified design concerns to estimate sample representation in a given dataset and comprehensibly explain sample selection bias.



(a) Sample-representation from SHAP scores **(b)** Marginal sample representation

Fig. 7: Explaining sample representation using simulation (F=VGG13, $T = T^*$, $S^T = 0.7$)

4 Discussion

Under-representation and outlier detection A good outlier detector $E_{\mathcal{S}}$ of under-represented samples must blur the distinction between simulated and real images while emphasizing the distinction between over and under-represented images. Figure 6b shows both conditions are jointly achievable, with classifiers that have a high test set accuracy, and therefore generalize well, also having better AUROC scores in detecting representation. However, as seen in Figure 8, using regularization measures like batch normalization layers after each convolutional block, while improving test accuracy, reduces AUROC scores for all classifier instances. This is probably because it tends to blur [23] both forms of distinction. The figure also shows that dropout increases the test accuracy without any major effect on AUROC scores, giving no special domain separation advantage in detecting under-representation. Among the classifier configurations investigated here, vanilla VGG, with the strongest correlation between AUROC and test set accuracy, is observed to best addresses both forms of domain distinction.

T	P	Y^1	$V^j(P)$					Mean $V^j(P)$
			j=2	3	4	5	6	
-	$P_{\mathcal{T}}$	0	0.60	0.47	0.59	0.47	0.75	0.57
		1	0.57	0.45	0.57	0.45	0.76	
T^* $S^T = 0.7$	$P^+_{\mathcal{T}}$ F =VGG13	0	0.49	0.14	0.17	0.35	0.55	0.27
		1	-0.19	0.26	0.16	0.17	0.56	
	$P^+_{\mathcal{T}}$ F =VGG05	0	0.47	0.33	0.29	0.44	0.60	0.39
		1	0.31	0.34	0.27	0.25	0.56	
1	$P^+_{\mathcal{T}}$ F =VGG13	0	0.49	0.30	0.40	0.15	0.69	0.36
		1	0.30	0.28	0.21	0.13	0.69	
	$P^+_{\mathcal{T}}$ F =VGG05	0	0.22	0.14	0.54	0.47	0.65	0.43
		1	0.57	0.29	0.56	0.15	0.70	

Table 1: Quantitative bias estimation

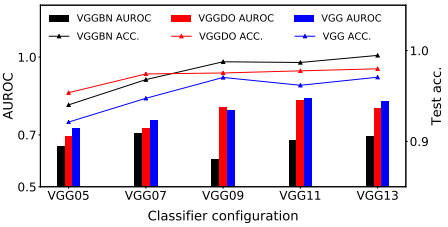


Fig. 8: Effect of regularization on AUROC

The importance of effective simulation It is crucial to note that high test accuracy reflects the combined effect of plausible simulation and good generalization. It is equally essential, therefore, that the simulator produces samples that are plausibly real. Ensuring effective simulation, while supporting a variety of parameters, is undoubtedly a challenge for realistic datasets with richer content. As noted earlier, while this is domain and problem dependent, for images at least, rapid advancements in the quality and range of graphics tools ([5],[9]), potentially makes effective simulation plausible. However, with notable progress in techniques that automate parts of the labeling process [7], it is also important to assess whether labeling is cheaper for the dataset in concern.

Improving estimation of representation Figure 7b shows that while the estimated sample representation $P^+_{\mathcal{T}}$ comes close, it does not overlap perfectly with the true label distribution $P_{\mathcal{S}}$. As quantified in Table 1, even the best detector (F=VGG13 at $T = T^*$) has a mean overlap index of 0.27 indicating relatively close, but only partial, overlap on average. At the individual label level, index values show varying accuracy in support-matching. The representation of pixel brightness $0.5 < V^6(P_{\mathcal{T}}) < 0.8$ is consistently underestimated, while those of bounding box coordinates are better estimated. It is however clear from Table 1 that temperature scaling ($T = T^*$ vs 1) and deeper classifiers (F =VGG13 vs VGG05) improve estimation, indicating that more sophisticated techniques of predictive outlier detection, like methods in [32], can improve estimation.

Balancing detail in specifying expectations The level of detail specified in the expectations $P_{\mathcal{T}}$ plays a key role in deciding the cost and benefit of explaining sample representation. An overly detailed breakdown of design factors involves significant engineering effort, degrades interpretability, and overlooks the remarkable benefits of generalization offered by deep learning. But well-balanced expectations can provide valuable insight into training data. Take an application like self-driving vehicles, where engineers actively seek a certain level of understanding of operational scenarios [15] to ensure safe operation. Such understanding can be exploited to systematically explain, analyze, and manage the data used to train models deployed in the system, thereby improving overall

confidence in its dependability. While balancing details in the specification may not always be easy, one advantage of this method is that it is semi-supervised. Annotations included in the analysis impacts only the simulated test set $\mathcal{T}$ and has no effect on the actual dataset $\mathcal{S}$.

Extension to other domains This method of explanation can conceivably be extended to a problem in another domain if (i) operational scenarios can be reasonably broken down and (ii) model-based parametric simulators that can generate data for this domain are available. For example, this method can use a simulator of vulnerable road user trajectories [16] to examine a sparsely labeled dataset of trajectories (e.g. [30]) and check whether it adequately represents trajectories of risk groups like elder pedestrians, electric bikes, etc.

5 Related work

Sample selection bias Sample selection bias has been addressed in existing literature from the perspective of domain adaptation [19]. Previous methods to mitigate sample selection bias have mainly attempted to modify the training procedures or the model itself to yield classifiers that work well on the test distribution. Methods such as importance re-weighting [35], minimax optimization [24], kernel density estimation [8] and model averaging [10] all fall in this category. While these methods can yield classifiers that are able to generalize, the accuracy can suffer when the two distributions differ greatly in the overlap of their support or in the distribution of their mass. Our immediate goal, on the other hand, does not seek to obtain a classifier that generalizes, but instead we seek to obtain a high level *understanding* of the deficiencies of our training data and where the bias stems from. This goal does not necessarily require a full specification of $P(Y)$, instead we work with the weak proxy of $P_{\mathcal{T}}(Y)$ which attempts to match $P(Y)$ only through the support. However, by eschewing mass-modeling, we gain a few advantages, one of which is the reduced effort in defining expectations. More importantly, since several existing methods for correcting sample selection bias work only if the support of $P_{\mathcal{T}}$ is included in that of $P_{\mathcal{S}}$ and our method of explanation tests precisely for this condition. Overlap indices $V^j(P_{\mathcal{T}}) \leq 0$ guarantees that the support of the biased distribution includes that of the expectations and correction measures like importance re-weighting are applicable. If $0 < V^j(P_{\mathcal{T}}) \leq 1$, expanding the diversity of data collection is unavoidable. Thus seeking to understand and explain the data set can allow for an improved understanding of the validity for methods that directly impacts the generalization performance.

Understanding sample representation Besides clustering approaches [6] and feature projection methods such as t-SNE [26], previous research into providing a high level understanding of the training set has, for example, applied tree-based methods to detect regions of low point density in the input space [13].

High-dimensional explanations in the input space, however, adversely affects interpretation, and ways to extend these methods to yield explanations using an interpretable low-dimensional space of annotations are not immediately clear.

Bias estimation using simulation Closer to our purpose are the methods [28] and [27] which detect inherent biases in a trained model using parametric simulation and Bayesian optimization. While their goal is to find input samples where the model is locally weak, our goal is to ensure that a given dataset meets global expectations defined by a test set. This can verify that a system is dependable for all considered scenarios, like [34], which is a standardized set of tests. However, in reformulating bias detection as outlier detection, our method – unlike the aforementioned methods – trades-off the ability to detect unknown unknowns [21] in favor of a faster, global evaluation of bias. Combining our global and their local approaches may, therefore, help ensure better overall dependability.

Shapley-based outlier detection Previous work using Shapley values for outlier detection, such as [12] and [1], focus mainly on providing interpretable explanations for why a data point is considered to be an outlier. It may also be possible to extend their data-space explanations to the annotation-space, like we do, using parametric simulation. However, pixel-wise reconstruction error has well-known drawbacks in capturing structural aspects of data [22]. It is therefore not immediately clear whether their use of auto-encoder reconstruction error is as good at detecting structural under-representation as our technique of using predictive certainty, which is calculated from the feature space of a classifier.

6 Conclusions

With data playing a crucial role in deciding the behavior of trained models, evaluating whether training and validation sets meets design expectations would be a helpful step towards a better understanding of model properties. To aid this evaluation, we demonstrate a method to specify expectations on and evaluate sample representation in a dataset, in a human interpretable form, in terms of annotations. Using parametric simulation to map test annotations into a test set, the method exposes under-representation by measuring the uncertainty of a classifier, trained on the original dataset, in recognizing test set samples. Techniques of input attribution enable further conversion of predictive uncertainty into a comprehensible low-dimensional estimate of sample representation in the dataset. While refinements in estimation are possible, the core quantitative and qualitative methods shown here are valuable aids in understanding a dataset and, consequently, the properties of a model trained using this data.

References

1. Antwarg, L., Shapira, B., Rokach, L.: Explaining anomalies detected by autoencoders using SHAP. CoRR **abs/1903.02407** (2019), http://arxiv.org/abs/1903.02407
2. Berman, D.S., Buczak, A.L., Chavis, J.S., Corbett, C.L.: A survey of deep learning methods for cyber security. Information **10**(4), 122 (2019). https://doi.org/10.3390/info10040122
3. Birch, J., Rivett, R., Habli, I., Bradshaw, B., Botham, J., Higham, D., Jesty, P., Monkhouse, H., Palin, R.: Safety cases and their role in ISO 26262 functional safety assessment. In: Bitsch, F., Guiochet, J., Kaâniche, M. (eds.) Computer Safety, Reliability, and Security - 32nd International Conference, SAFECOMP 2013, Toulouse, France, September 24-27, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8153, pp. 154–165. Springer (2013). https://doi.org/10.1007/978-3-642-40793-2_15
4. Borg, M., Englund, C., Wnuk, K., Durán, B., Levandowski, C., Gao, S., Tan, Y., Kaijser, H., Lönn, H., Törnqvist, J.: Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. CoRR **abs/1812.05389** (2018), http://arxiv.org/abs/1812.05389
5. Chao, Q., Bi, H., Li, W., Mao, T., Wang, Z., Lin, M.C., Deng, Z.: A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. Comput. Graph. Forum **39**(1), 287–308 (2020). https://doi.org/10.1111/cgf.13803
6. Chen, J., Chang, Y., Hobbs, B., Castaldi, P.J., Cho, M.H., Silverman, E.K., Dy, J.G.: Interpretable clustering via discriminative rectangle mixture model. In: Bonchi, F., Domingo-Ferrer, J., Baeza-Yates, R., Zhou, Z., Wu, X. (eds.) IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain. pp. 823–828. IEEE Computer Society (2016). https://doi.org/10.1109/ICDM.2016.0097
7. Cheng, Q., Zhang, Q., Fu, P., Tu, C., Li, S.: A survey and analysis on automatic image annotation. Pattern Recognit. **79**, 242–259 (2018). https://doi.org/10.1016/j.patcog.2018.02.017
8. Dudík, M., Schapire, R.E., Phillips, S.J.: Correcting sample selection bias in maximum entropy density estimation. In: Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]. pp. 323–330 (2005), http://papers.nips.cc/paper/2929-correcting-sample-selection-bias-in-maximum-entropy-density-estimation
9. Ersotelos, N., Dong, F.: Building highly realistic facial modeling and animation: a survey. The Visual Computer **24**(1), 13–30 (2008). https://doi.org/10.1007/s00371-007-0175-y
10. Fan, W., Davidson, I.: On sample selection bias and its efficient correction via model averaging and unlabeled examples. In: Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA. pp. 320–331. SIAM (2007). https://doi.org/10.1137/1.9781611972771.29, https://doi.org/10.1137/1.9781611972771.29
11. Gardner, A., Kanno, J., Duncan, C.A., Selmic, R.R.: Measuring distance between unordered sets of different sizes. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. pp. 137–143. IEEE Computer Society (2014). https://doi.org/10.1109/CVPR.2014.25

12. Giurgiu, I., Schumann, A.: Additive explanations for anomalies detected from multivariate temporal data. In: Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E.A., Carmel, D., He, Q., Yu, J.X. (eds.) Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019. pp. 2245–2248. ACM (2019). https://doi.org/10.1145/3357384.3358121
13. Gu, X., Easwaran, A.: Towards safe machine learning for CPS: infer uncertainty from training data. In: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2019, Montreal, QC, Canada, April 16-18, 2019. pp. 249–258. ACM (2019). https://doi.org/10.1145/3302509.3311038
14. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1321–1330 (2017), http://proceedings.mlr.press/v70/guo17a.html
15. Gyllenhammar, M., Johansson, R., Warg, F., Chen, D., Heyn, H.M., Sanfridson, M., Söderberg, J., Thorsén, A., Ursing, S.: Towards an Operational Design Domain That Supports the Safety Argumentation of an Automated Driving System. In: 10th European Congress on Embedded Real Time Software and Systems (ERTS 2020). TOULOUSE, France (Jan 2020), https://hal.archives-ouvertes.fr/hal-02456077
16. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Physical Review E **51**(5), 4282–4286 (May 1995). https://doi.org/10.1103/physreve.51.4282, http://dx.doi.org/10.1103/PhysRevE.51.4282
17. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. CoRR **abs/1610.02136** (2016), http://arxiv.org/abs/1610.02136
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
19. Kouw, W.M.: An introduction to domain adaptation and transfer learning. CoRR **abs/1812.11806** (2018), http://arxiv.org/abs/1812.11806
20. Kuutti, S., Bowden, R., Jin, Y., Barber, P., Fallah, S.: A survey of deep learning applications to autonomous vehicle control. CoRR **abs/1912.10773** (2019), http://arxiv.org/abs/1912.10773
21. Lakkaraju, H., Kamar, E., Caruana, R., Horvitz, E.: In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 2124–2132. AAAI Press (2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14434
22. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. pp. 1558–1566 (2016), http://proceedings.mlr.press/v48/larsen16.html
23. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=Hk6dkJQFx
24. Liu, A., Ziebart, B.D.: Robust classification under sample selection bias. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014,

Montreal, Quebec, Canada. pp. 37–45 (2014), http://papers.nips.cc/paper/5458-robust-classification-under-sample-selection-bias
25. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 4765–4774 (2017), http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions
26. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008), http://www.jmlr.org/papers/v9/vandermaaten08a.html
27. McDuff, D.J., Cheng, R., Kapoor, A.: Identifying bias in AI using simulation. CoRR **abs/1810.00471** (2018), http://arxiv.org/abs/1810.00471
28. McDuff, D.J., Ma, S., Song, Y., Kapoor, A.: Characterizing bias in classifiers using generative models (2019), http://papers.nips.cc/paper/8780-characterizing-bias-in-classifiers-using-generative-models
29. Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., Mordvintsev, A.: The building blocks of interpretability. Distill (2018), https://distill.pub/2018/building-blocks/
30. Rasouli, A., Kotseruba, I., Kunic, T., Tsotsos, J.K.: PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 6261–6270. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00636, https://doi.org/10.1109/ICCV.2019.00636
31. Seshia, S.A., Sadigh, D.: Towards verified artificial intelligence. CoRR **abs/1606.08514** (2016), http://arxiv.org/abs/1606.08514
32. Shafaei, A., Schmidt, M., Little, J.J.: A less biased evaluation of out-of-distribution sample detectors. In: 30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019. p. 3. BMVA Press (2019), https://bmvc2019.org/wp-content/uploads/papers/0333-paper.pdf
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1409.1556
34. Thorn, E., Kimmel, S.C., Chaka, M.: chap. A Framework for Automated Driving System Testable Cases and Scenarios (Sep 2018), https://rosap.ntl.bts.gov/view/dot/38824, tech Report
35. Tran, V.: Selection Bias Correction in Supervised Learning with Importance Weight. (L'apprentissage des modèles graphiques probabilistes et la correction de biais sélection). Ph.D. thesis, University of Lyon, France (2017), https://tel.archives-ouvertes.fr/tel-01661470
36. Vogelsang, A., Borg, M.: Requirements engineering for machine learning: Perspectives from data scientists. In: 27th IEEE International Requirements Engineering Conference Workshops, RE 2019 Workshops, Jeju Island, Korea (South), September 23-27, 2019. pp. 245–251. IEEE (2019). https://doi.org/10.1109/REW.2019.00050
37. Zadrozny, B.: Learning and evaluating classifiers under sample selection bias. In: Brodley, C.E. (ed.) Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004. ACM International Conference Proceeding Series, vol. 69. ACM (2004). https://doi.org/10.1145/1015330.1015425

Latent Causation: An algorithm for pairs of correlated latent variables in Linear Non-Gaussian Structural Equation Modeling*

Arnaud Pollaris and Gianluca Bontempi

Université Libre de Bruxelles, ULB CP212, boulevard du Triomphe, 1050 Bruxelles, Belgium Arnaud.Pollaris@ulb.ac.be and Gianluca.Bontempi@ulb.ac.be
https://mlg.ulb.ac.be

Abstract. This paper addresses the problem of inferring causation in a pair of linearly correlated continuous latent variables. We first discuss the limitations of the Direction Dependance Analysis (DDA) approach and then introduce the Latent Causation (LC). Five variants (in terms of dependency statistic) of the LC algorithm are assessed with ROC curves, then we consider the case of a latent confounder (uniform or chi-square distributed). While the distribution and the correlations of the latent confounder influence the accuracy, experimental results show the robustness of the method using bootstrapped p-values. Implications and limits of the experimental results are then discussed together with future directions.

Keywords: SEM · Latent Variables · Causal inference · Observational data · Latent Confounder · Non normality · Simulations.

1 Introduction

An observed dependency between two variables A and B may have four different explanations assuming no feedback loop: (1) A is a (direct or indirect) cause of B, (2) B is a (direct or indirect) cause of A, (3) there is a hidden common ancestor U of A and B, (4) a common descendant of A and B is kept fixed in the observed dataset.

This paper lies at the crossroad between Structural Equation Modeling (SEM) and causal inference literature. Widely used in psychology and in management research, SEM is a family of techniques which allows the analysis of relationships between continuous latent variables. Whereas the capacity of SEM to support causal inference has been discussed during decades (Bollen and Pearl, 2013), we consider here classical SEM as a set of confirmatory techniques since the causal graph specified by the user can (and should) be drawn before the data collection. In that perspective, the fitting of the data on one or more competing causal models should allow to reject wrong models and then inform the scientist that at least one of its related causal assumptions is wrong (see Bollen and Pearl,

* We would like to thank the anonymous reviewers for their helpful comments.

2013). However, for equivalent models (i.e., *alternative models that fit any data to the same degree* (MacCallum & Austin, 2000 p. 213)) it is not always easy to retrieve information about causation. So additional tools are needed.

In machine learning, causal discovery is not confirmatory but exploratory: its goal is to build a causal model based on available data (data collection comes before the learning of a causal graph). Though in causal discovery most algorithms make the assumption of causal sufficiency (also in cause-effect pairs (e.g., Guyon, 2014)), some of them address latent variables, e.g., the BPC algorithm (Silva, Scheine, Glymour, and Spirtes, 2006), the FindOneFactorClusters (FOFC) algorithm (Kummerfeld and Ramsey, 2016) or, more recently, the LSTC algorithm (Cai, Xie, Glymour, Hao & Zhang, 2019). Shimizu, Hoyer and Hyvärinen (2009) also show that a linear acyclic model for latent factors is identifiable when the data are non normal. However, our work in this paper differs from this literature because we assume the structure of the measurement model already known (i.e.: each indicator has been specified as measuring exactly one latent variable of interest in our models) and we do not focus on building large causal graphs from data.

In particular we focus on causal inference in pairs of latent variables. In a confirmatory perspective, assuming linearity and non normality, the Direction Dependent Analysis project (DDA project, 2020) offers an interesting starting point to infer causation in a pair of latent variables since indications for a latent confounder can also be detected using its independence component. However, as stressed in (Wiedermann, Merkle, & von Eye, 2018), there is still a need for improving the trustworthiness of the DDA approach in presence of meaningful confounding. For this reason, in this paper we focus on improving the independence component of the DDA approach by focusing in particular on discriminating between causal and spurious confounding latent configurations.

The paper is structured as follows: First, we present the causal inference setting we are interested in. Next, the DDA approach is introduced. Then, limitations for using DDA with latent variables are presented. Next, we propose the Latent Causation (LC) algorithm, grounded on the third DDA component "Independence properties of predictor and error term" (see Wiedermann & Li, 2018). Then, we present some experiments on simulated data: benchmarking LC with respect to to state-of-the-art DDA and sensitivity study of LC.

2 Problem setting

Let us consider two continuous correlated latent variables, denoted ξ and η and some observable children variables called "indicators" (e.g., Kline, 2011) which are functions of a latent variable plus an additive independent noise. Figure 1 visualizes a causal and a confounding topology we want to discriminate between. As an example, values and distributions specified in Figure 1 are possible instances which are used below as assignations for parameters in our simulations. The number of indicators can also differ from instances in Figure 1. While we want to confirm the correct causal direction $\xi \rightarrow \eta$ (and not $\xi \leftarrow \eta$) in causal

models like Model 1, we also want to make sure we will not conclude in favor of a causal direction for pairs (ξ, η) only correlated due to a latent confounder (called U below) like in Model 0. Throughout the paper, linearity is assumed, variables are continuous and all coefficients in theoretical models are presented for standardized variables.

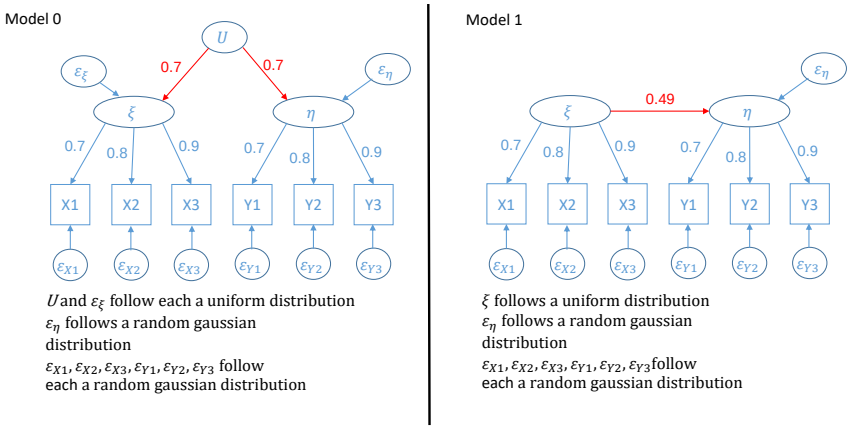


Fig. 1. Latent variables are represented by ellipses or circles. Observed variables are represented by rectangular boxes. Both models are completely standardized. In Model 1, there is a causation between ξ and η but not in Model 0. Note that the distributions of ξ and η also differ between both models.

3 The DDA approach

The DDA project regroups the techniques which address the inference of causation by considering 3 aspects: 1. distributional properties of observed variables, 2. distributional properties of error terms of competing models and 3. independence properties of predictor and error terms of competing models. In this paper we will focus on the third aspect. The rationale of our work resides in this consideration by Wiedermann & Sebastian (2019a, p. 15): *"Considering the behavior of DDA components under confounding, the DDA independence component is the most important criterion to confirm that no strong confounders are present (or, at least, that the influence of confounders is minimal). Thus, HSIC- and dCor-tests are crucial DDA procedures. When these tests indicate the presence of meaningful confounding, results of the remaining DDA procedures are no longer trustworthy."*

The algorithm used for the (third) DDA independence component is already well-established in machine learning (see Peters, Janzing & Schölkopf, 2018, p. 62):

1. Fit a regression model $\hat{f}_Y$ of Y on X
2. Test whether the residual $Y - \hat{f}_Y(X)$ is independent of X.
3. Repeat the procedure by exchanging the roles of X and Y.
4. If the independence is accepted for one direction and rejected for the other, infer the former one as the causal direction.

However, while the algorithm presented by Peters et al. (2018) is commonly used to determine the causal direction for data with relationships strictly non-linear, in the DDA framework, this algorithm is used assuming linearity and non-normality (Thoemmes, 2019).

Here is the rationale behind the algorithm for DDA. For instance, let us then assume that X and Y are two related continuous random variables such that :

$$Y = aX + \epsilon_Y \qquad \text{with } a \neq 0 \tag{1}$$

where ϵ_Y is the error term from a regression where Y is explained linearly as a function of X. And let us assume that either X or ϵ_Y is not normally distributed. In this context, we get the falsehood of the expression

$$X \perp\!\!\!\perp \epsilon_Y \qquad \text{AND} \qquad Y \perp\!\!\!\perp \epsilon_X \tag{2}$$

where ϵ_X is the error term from an alternative linear regression having X explained linearly as a function of Y:

$$\epsilon_X = X - bY \qquad \text{with } b \neq 0 \tag{3}$$

The above reasoning relies on the corollary of the Darmois-Skitovich theorem (see e.g.: Eberhardt, 2017, p.86):

Corollary 1. *Let $X_1, ..., X_n$ be independent, non-degenerate random variables. If for two linear combinations :*

$$l_1 = a_1X_1 + ... + a_nX_n \qquad \textit{with } a_i \neq 0 \tag{4}$$

$$l_2 = b_1X_1 + ... + b_nX_n \qquad \textit{with } b_i \neq 0 \tag{5}$$

at least one X_i is not normally distributed, then l_1 and l_2 are not independent.

After substitution of Y in (3) by its expression from (1):

$$\epsilon_X = X - b(aX + \epsilon_Y) = (1 - ab)X - b\ \epsilon_Y \tag{6}$$

it appears both ϵ_X and Y are linear combinations of X and ϵ_Y. Applying Corollary 1, it can then be affirmed that if X and ϵ_Y are independent, non-degenerate random variables that are not normally distributed in (1), then Y and ϵ_X can not be independent in (3).

Then, as nicely illustrated in Spirtes & Zhang (2016) and in Wiedermann & Li (2018), this asymmetric pattern of the causality can leave a footprint in the data. Furthermore, the shape of the distribution does not matter, since it is not a normal distribution.

3.1 Limitations of the current DDA approach

Here we address the limitations of the current DDA approach for causal inference in pairs of latent variables:

1. **DDA does not exploit all the available information in measures of dependencies**. The DDA independence component approach relies on a combination of two statistical tests of independence (see e.g., in equation (2), one test for each possible causal direction). Four conclusions are possible: (a) rejection of both independences (i.e., suspicion of confounder), (b) no rejection of both independences, and rejection of only one of the two independences which gives either (c) $X \rightarrow Y$ or (d) $Y \rightarrow X$. It is worth remarking what follows:
   (a) **A non-significant result for a test of independence is recommended** to infer a causal direction. But, the lack of independence rejection is not a proof of independence. So, in DDA, how to be sure we are not missing a confounder because of a lack of power in the test?
   (b) **Insufficient use of the continuum of dependencies.** Given a pair of variables, there is not only simplistically either "independence" or "dependence" but a whole range of dependence strengths. Even if DDA concludes in favour of the presence of a confounder (i.e, both independences are rejected), the comparison of statistics of dependence from both tests may convey additional information to favour one of the two causal directions, under the assumption there is also causation in the pair of variables of interest (beyond a simple spurious correlation).
2. **Limit of the DDA distinction between "presence of a confounder" and "causation".** In the DDA framework, the presence of a confounder may be revealed by rejection of the independence for both directions whereas a causation should be revealed by the rejection of the independence for only one direction. But some theoretical models can include both causation and confounder. Considering pairs of latent variables like in the example of the causal Model 1 (Fig. 1, right) with no confounder between ξ and η, the latent variable ξ can also be considered itself as a latent confounder between the two groups of observed indicators: $(X1, X2, X3)$ and $(Y1, Y2, Y3)$, where each indicator is a linear combination including the latent variable ξ. So, since ξ can be both a cause of η and a confounder between the two groups of indicators, maybe it would be better not to rely on the DDA independence component if we want to confirm there is no additional confounder between latent ξ and latent η when it concludes in favor of a causation. And, if the DDA independence component concludes there is a confounder, the question of the direction for a possible causation remains still open after.

So, to infer a causation in a pair of non-Gaussian linearly related latent variables, the point is maybe not to make sure first there is no latent confounder but to focus instead on a direct comparison of the strength of each dependence (one for each possible causal direction) to try to infer (if possible and with an associated level of confidence) a causal direction like LC does.

4 The Latent Causation algorithm

The main difference between the LC and the DDA independence component is related to the computation of differences between the statistics of dependency, which is an essential element to identify a causal direction. The Latent Causation (LC) pseudocode is detailed below. After the computation of factor scores (F_ξ, F_η) in steps (2a) and (2b) representing latent variables ξ and η respectively, the steps (2c), (2d), (2e) and (2f) implement the DDA independence component. However, if there is a true causal link between ξ and η, a difference of dependence should be observed between values computed in (2e) and (2f). Unlike classical DDA (yet inspired to a sensitivity analysis performed by Wiedermann and Sebastian (2019b) using bootstrap), the difference of values in (2f) and (2e) is always saved by LC in step (2g). While a positive score in (2g) is favouring one causal direction, a negative score is favouring the opposite causal direction. A non-parametric bootstrap (B resamples) is then adopted to assess the significativity of the consensus. LC may reach 3 conclusions in step (4): "infer $\eta \to \xi$", "infer $\xi \to \eta$" or "data do not allow to conclude." Unlike in classical DDA, we do not need to have a non significant p-value to conclude for a causal direction.

The Latent Causation algorithm

Input:

- An observed dataset with indicators divided in 2 pre-defined groups (with no overlap): $\mathbf{X}$ for the indicators of ξ, $\mathbf{Y}$ for the indicators of η.
- A metric to rate the strength of a bivariate dependence
- α: a threshold (to define acceptable type I error rate)
- B: number of bootstrap datasets

Output: A decision taken by the algorithm:
"infer $\eta \to \xi$" OR "infer $\xi \to \eta$" OR "data do not allow to conclude."

1. From the original sample of size n, draw B bootstrap samples ($size = n$, with replacement).
2. For each bootstrapped sample do :
   (a) Compute the factor scores "F_ξ" to represent ξ using $\mathbf{X}$ (exclusively)
   (b) Compute the factor scores "F_η" to represent η using $\mathbf{Y}$ (exclusively)
   (c) Regress linearly F_η *as a function of* F_ξ and save the residuals ($resid_{F_\eta}$)
   (d) Regress linearly F_ξ *as a function of* F_η and save the residuals ($resid_{F_\xi}$)
   (e) Measure how strong $dependence(resid_{F_\eta}, F_\xi)$ is
   (f) Measure how strong $dependence(resid_{F_\xi}, F_\eta)$ is
   (g) Save the difference between both measures from (f) and (e)
3. Based on the B saved differences (in 2g), select a percentile confidence interval based on probabilities ($\alpha/2$; $1 - \alpha/2$).
4. Select a conclusion:
   - If 0 is not included in the confidence interval:
     - If a majority of bootrapped samples gave:
       $dependence(resid_{F_\eta}, F_\xi) > dependence(resid_{F_\xi}, F_\eta)$:
       "infer $\eta \to \xi$"

    - Else:
      "infer $\xi \rightarrow \eta$"
  - Else:
    "data do not allow to conclude."

Some considerations on the LC algorithms follow:

- Factor scores computation: since the information about the latent variables ξ and η is only available through noisy indicators, the question about their representation naturally arises. While each indicator is assumed to be a linear descendant of a specific latent variable of interest, we choose Principal Component Analysis (PCA) (Husson, Lê & Pagès, 2009) to compute factor scores separately for each latent variable ξ and η.
- Dependency measures: we considered 5 measures in our LC experiments:
  - Spearman's correlation (in absolute value).
  - Brownian distance correlation (Szekely, Rizzo, & Bakirov, 2007): it returns the dCor statistic (a score between [0;1] where 0 stands for independence).
  - dCor's p-value : it estimates a p-value using permutation bootstrap.
  - Hilbert-Schmidt Independence Criterion (HSIC; Gretton, Fukumizu, Teo, Song, Schölkopf & Smola, 2008): The closer HSIC is with 0, the weaker is the dependence.
  - HSIC's gamma-approximated p-value: The smaller the p-value is, the more we are in independence rejection.
- LC relies on some assumptions:
  - Two correlated non normal latent variables (i.e., ξ and η).
  - All the relationships are linear (measurement model included).
  - There is no cycle in the causations.
  - Two distinct groups of indicators are available for ξ and for η respectively. Each indicator is strongly correlated (e.g., Pearson's correlation ≥ 0.7 in our simulations) with its corresponding latent variable (either ξ or η).
  - Each indicator is linearly function of its latent variable + an independent random Gaussian noise.
  - If there is a causal effect between ξ and η, it is assumed the effect is the same for every individual (causal effect homogeneity).
  - There is no unusual or influential observations.
  - All the variables are continuous.

5 Experimental results

The experimental results are divided in two subsections: first, we use simulations to compare LC and DDA. Second, additional analysis are provided to further explore the performances of five LC variants.

5.1 Benchmarking LC vs DDA

Data generation. In this section, to compare LC and DDA, we used the causal structures from Model 0 and Model 1 to generate datasets by Monte Carlo (1000 datasets for each model). The distributions and the standardized values specified for the different coefficients (at the Population level) are available in Fig. 1. Since we want to infer causation beyond correlation, we arbitrary specify in every simulation in this paper the same theoretical Pearson's correlation of 0.49 between ξ and η.

To generate our datasets and compare the methods, we implemented a simulator in the R language (R Core Team, 2019)[1]. We consider two groups of 3 indicators: X_1, X_2, X_3 and Y_1, Y_2, Y_3 measuring latent ξ and η respectively. Since we cannot directly apply the DDA independence component on observed indicators, factor scores representing ξ and η are first computed using the first axis in two separated PCA (i.e., first axis build on X_1, X_2, X_3 and other first axis build on Y_1, Y_2, Y_3). So, we used PCA[2] in a similar way in LC and before applying DDA.

Accuracy assessment. Table 1 reports the comparison DDA vs LC in terms of accuracy. In DDA, since two p-values are used for taking decisions, 4 conclusions are possible: ξ causes η, η causes ξ, suspicion of a latent confounder (both p-values are sig.) and the "do not conclude" option (none of the 2 p-values is sig.). In LC, based on a bootstrapped confidence interval, 3 conclusions are possible: ξ causes η, η causes ξ and the "do not conclude" option.

Concerning DDA for Model 0 using HSIC gamma p-value or HSIC p-value bootstrap (Sen and Sen, 2014) as independence test, it appears that $(72 + 43)/1000 = 11.5\%$ and $(100 + 84)/1000 = 18.4\%$ of the conclusions are false positives (FP) (indicating wrongly causation) which exceeds in both cases the maximum $\alpha = 5\%$ allowed. In contrast, DDA's dCor p-values seem to work fine on Model 0 (FP rate: 0.4%), though this method shows less statistical power (only 303 on 1000 datasets were true positive (TP) causal conclusions) than LC variants (401 TP using Spearman as independence statistic, 510 TP using dCor-stat, 343 TP using dCor-p-value, 594 TP using HSIC-stat and 608 TP using HSIC's gamma-approximated p-value).

In Table 1, all LC variants show a FP rate under the expected $\alpha = 5\%$ (e.g., the observed total FP rate using HSIC-stat $= (1 + 2)/1000 = 0.3\%$) and outperform in power (i.e., number of TP in Model 1) DDA's dCor which is the only considered DDA variant with a FP rate below $\alpha = 5\%$.

Discussion of results. To infer causation in a pair of correlated latent variables, we are looking for an algorithm with low type I error rate (i.e a proportion of FP below the specified value for α) when there is no true causation between ξ

[1] Code in https://github.com/apollaris/LatentCausation

[2] In experiments we use the PCA function (using default option "scale.unit = TRUE") from the R package FactoMineR (Le, Josse & Husson, 2008).

Table 1. The DDA independence component applied on factor scores from PCAs vs. the LC algorithm (DNC stands for "Do not conclude")

DDA independence component	**Model 1:** true causation $\xi \rightarrow \eta$				**Model 0:** no causation but a latent confounder			
Independence statistic	$\xi \rightarrow \eta$ (TP)	$\eta \rightarrow \xi$ (FN)	Confounder (FN)	DNC (FN)	$\xi \rightarrow \eta$ (FP)	$\eta \rightarrow \xi$ (FP)	Confounder (TN)	DNC (TN)
dCor	303	0	0	697	2	2	0	996
HSIC - gamma	860	1	28	111	72	43	34	851
HSIC - bootstrap	870	1	71	58	100	84	62	754
LC algorithm	**Model 1:** true causation $\xi \rightarrow \eta$				**Model 0:** no causation but a latent confounder			
Independence statistic	$\xi \rightarrow \eta$ (TP)	$\eta \rightarrow \xi$ (FN)	DNC (FN)		$\xi \rightarrow \eta$ (FP)	$\eta \rightarrow \xi$ (FP)	DNC (TN)	
Spearman	401	0	599		19	0	981	
dCor - stat	510	0	490		1	0	999	
dCor - p-value	343	0	657		0	0	1000	
HSIC - stat	594	0	406		1	2	997	
HSIC - p-value gamma	608	0	392		0	3	997	

Parameters specification:
- **For both DDA and LC:** 1000 samples generated for Model 1 and 1000 samples for Model 0; sample size=500; α=0.05
- **For DDA only:** the number of replicates used for the estimation of each dCor-pvalue and the number of resamples used to compute each bootstrap's HSIC p-value were both set equal to 500.
- **For LC only:**, we used $B = 1000$ (bootstrap datasets) and the number of replicates used for the estimation of each dCor-pvalue was always set equal to 300.

and η (e.g., Model 0) and with good ability to retrieve the correct causal direction (i.e., statistical power represented here by the number of TP for datasets from Model 1). Looking only in our results at methods with an acceptable type I error rate, it appears all the five variants of LC are more powerful than the only acceptable DDA variant (using dCor as independence statistic). Furthermore, Table 1 shows the impact of the dependence statistic on the results (see e.g., for LC: 343 TP using dCor - p-value but 608 TP using HSIC's gamma-approximated p-value). The next section will present a LC sensitivity study to assess the role of the five statistics.

5.2 LC sensitivity study

Here we perform additional simulations to study the sensitivity of the LC accuracy to its parameters. Good accuracy means low type I errors (i.e., the proportion of FP can not be larger than α in the absence of causation between ξ and η) and good statistical power (i.e., a large number of TP is expected under causation between ξ and η). First, we show the statistical power of LC increases for larger sample sizes. Next, α's value is manipulated to show that LC does not exceed the allowed the type I error rate. Then we show that the TP rate increases with α. Finally, ROC curves visualize the ability of the five LC variants to discriminate beween a spurious correlation (i.e., Model 0) and a causation (i.e., Model 1). Because many different confounders can make a pair of latent variable (ξ,η) correlate, we conclude this section by a robustness anal-

ysis to answer the question: "Can a latent confounder U (due to its distribution and its correlations with ξ and η) increase LC's number of FP in the absence of causation between ξ and η?"

Sample size can increase the power of LC. 1000 datasets of different sample sizes ($n = 200$, $n = 300$, $n = 400$, $n = 500$) were simulated according to Model 1 (Fig. 1, right), i.e. with a true latent causation $\xi \to \eta$.

As observed in Fig. 2 upper left, the larger the sample size, the better is LC in retrieving $\xi \to \eta$. Furthermore, comparing the five variants of independence statistics, the methods based on HSIC appear here to be the most powerful.

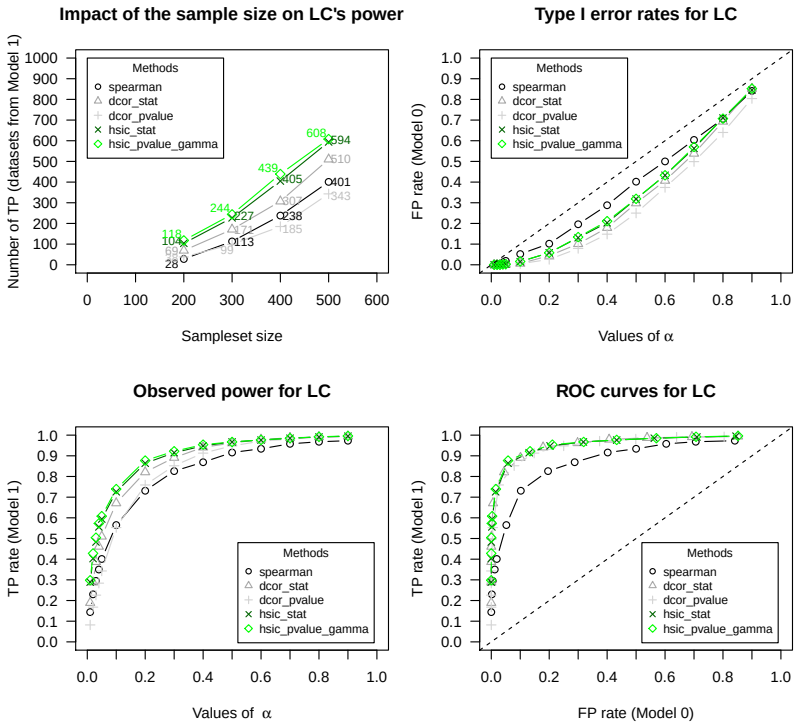


Fig. 2. Upper left: Power of the LC algorithm for retrieving the true $\xi \to \eta$ as a function of the sample size and the statistic used to measure independence; 1000 simulated datasets based on Model 1 for each sample size. Upper right: observed type I error rate (% of FP) for LC as a function of α. Lower left: observed power rate (% of TP) for LC as a function of α. Lower right: ROC curves for the 5 variants to measure dependence using LC ; for each curve, the different points correspond to different values assigned to α. For each last 3 plots: Sample size=500, 1000 simulated datasets for each estimation.

Manipulations on α. Using 1000 datasets generated from each model (Model 0 and Model 1) with a sample size $n = 500$, we manipulate the specified value of α to get the number of FP (for data generated from Model 0) and the number of TP (for data generated from Model 1). Here are the different values assigned to α: .01, .02, .03, .04, .05, .1, .2, .3, .4, .5, .6, .7, .8, .9.

Results in Fig. 2, upper right, show as expected for Model 0 that observed type I error rate (i.e., the proportion of FP) is always lower than the specified value of α in our simulations (whatever the method or the value of α).

Fig. 2 lower left shows that the number of correct causal direction (TP) increases with higher values of α and that the number of TP also differs between the five variants (measures of dependence). Notably, using HSIC's gamma-approximated p-values seem to get more TP compared with other observed methods. Then, by increasing α the number of both TP and FP increases as well. In Fig. 2 lower right, ROC curves to discriminate between Model 0 and Model 1 show that the methods (apart from Spearman correlation) present similar good abilities to discriminate between Model 1 (causation) and Model 0 (confounder).

Robustness: distribution and correlation with a latent confounder. Since estimated scores F_ξ, $resid_{F_\eta}$, F_η and $resid_{F_\xi}$ in Model 0 are all linear combinations of a sum of terms including the latent confounder U, according to Corollary 1 (of the Darmois-Skitovich theorem), $F_\xi \not\perp\!\!\!\perp resid_{F_\eta}$ and $F_\eta \not\perp\!\!\!\perp resid_{F_\xi}$ are expected together because U is not normally distributed (see also, Wiedermann et al., 2018). So, using additional simulated datasets (Monte Carlo), an empirical analysis of the robustness for LC is now performed to know if inflated type I errors (i.e, FP) can be avoided. While keeping constant the theoretical Pearson's correlation between ξ and η (as a reminder it was arbitrary set equal to 0.49 for our simulations), we generate additional datasets after manipulation of the distribution (symmetric uniform VS asymmetric chi-square) and the Pearson's correlation between U and ξ (and then also the Pearson's correlation between U and η) (see Fig. 3: Model 0 and its 4 variants : 0a, 0b, 0c and 0d for correlations assigned to the confounder). Because all methods to measure dependence were also compared here, we have now a "2 distributions of U $\times$ 5 models $\times$ 5 methods for statistics of dependence" design.

In the different plots in Fig. 4, the correlations of U influence the number of FP: for a strong correlation between U and ξ, the risk to conclude wrongly that ξ causes η increases ; on the opposite, when U mainly correlates with η, the risk to conclude wrongly that η causes ξ increases. However, the impact of the correlation of U is strongly reduced when U is symmetrical uniform (plots on the left) compared with an asymmetrical chi-square U ($df = 1$) (plots on the right). Fortunately, the problem of the distribution and correlations of U seems possible to overcome: looking at the dCor p-value method, the number of FP never exceeds 60, even when a very strong correlation of 0.875 between a chi-square U ($df = 1$) and η has been specified.

The last results seem to favour the method based of differences of dCor's p-values under the assumption of an influent latent confounder. However, a deeper

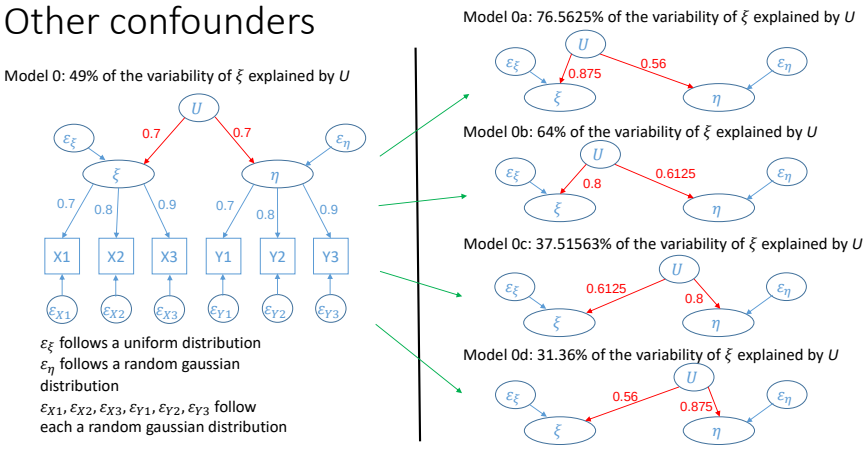


Fig. 3. In order to study the impact of the correlations of the confounder, simulations rely on variations of Model 0. While the population level's correlation between ξ and η is kept constant and equal to an arbitrary set value of 0.49, the confounder U is made more correlated with ξ in Models 0a and 0b and more correlated with η in Models 0c and 0d. In Models 0a, 0b, 0c, 0d, the measurement model (same as in Model 0) is not displayed to save space.

look at the results from extreme Models 0a and 0d with a chi-square U, reveals that for the variant using differences of HSIC's p-values, despite of the very high number of FP, the median of differences scores (-4.77e-3 for simulations based on Model 0a ; 2.14e-2 for simulations based on Model 0d) is not far from the expected 0. So, methods based on differences of p-values might have a small recurrent bias due to the presence of a (very) asymmetric, strongly correlated U with either ξ or η. Using difference of dCor's p-values, this bias can be hidden due to the random bootstrapped estimation of each p-value.

6 Conclusions and future directions

In this paper, we propose LC, an algorithm for causal inference in a pair of latent variables for confirmatory analysis. In this specific context, LC appears to be better suited than classical DDA to differentiate causation and confounder patterns from data. The resulting recommendation is then to enrich DDA analysis with bootstrapped differences of independence statistics (possibly also outside the context of latent variables).

Directions for LC improvement may also be considered. For instance, promising research directions to extend the current work are:

– **Considering alternative ways to compute factors scores.** There are indeed alternatives to PCAs to represent latent ξ and η.

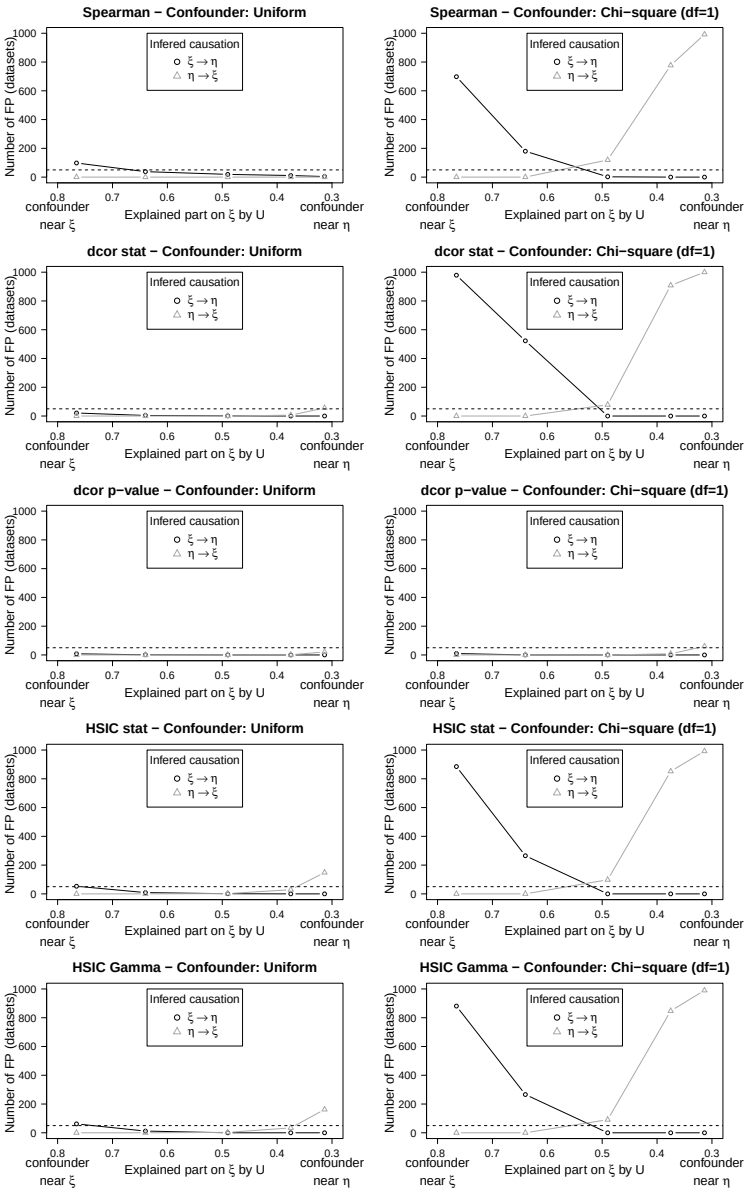


Fig. 4. Kinds of type I errors (FP) for each variant to measure dependence as a function of the distribution and the correlations of the latent confounder U (see Model 0 and variants 0a, 0b, 0c, 0d). Theoretical Pearson's correlation between ξ and η at the population level is always set equal to 0.49 in our simulation; sample size = 500, 1000 simulated datasets for each ten couples (5 models × 2 distributions of confounder)

- **Inclusion of an additional parameter allowing the user of LC to round to 0 each absolute difference of HSIC's p-values below a given specified threshold.** Whereas the variant based on differences of HSIC's gamma-approximated p-values gives some very good results, differences of bootstrap approximated dCor's p-values show more robustness in the presence of some specific latent confounders. However, about HSIC's gamma-approximated p-values, we remark that wrong conclusions in the robustness analysis come with a small persistent bias in the difference of p-values. So, rounding to 0 the very small observed differences in HSIC's gamma approximated p-value could improve its robustness.
- **Relaxing some assumptions of LC and comparison with other algorithms.** For instance, relaxing some assumptions, LC could be compared with parts of Cai et al. (2019)'s method. More widely speaking, future works should study the performances of LC under relaxed assumptions.
- **Manipulation of other parameters and additional comparisons using other models.** For instances, in our models, some distributions could be changed and other models could be considered. For instance, whereas Model 0 and Model 1 differed by the causal pattern and by the marginal distributions assigned to latent ξ and η respectively, an alternative for Model 0 would be to set exactly the same distributions for ξ and η than in Model 1 but with a specified theoretical correlation of 0. Furthermore, other models should also include confounder and causation together.
- **Presence of an observed confounder.** Corrections in each bootstrapped dataset could be included in LC to increase the accuracy.

Last but not least, future work should also test LC on real data as benchmark.

References

1. Bollen K. A. & Pearl, J.: Eight Myths about Causality and Structural Equation Models. In S.L. Morgan (Ed.), Handbook of Causal Analysis for Social Research, Chapter 15, 301-328. Springer. (2013) www.springer.com/lncs. Last accessed 24 Aug 2020
2. Cai, R., Xie, F., Glymour, C., Hao, Z., & Zhang, K.: Triad Constraints for Learning Causal Structure of Latent Variables. NeurIPS. (2019)
3. DDA Project Homepage, https://www.ddaproject.com/. Last accessed 14 July 2020
4. Eberhardt, F.: Introduction to the foundations of causal discovery. International Journal of Data Science and Analytics, **3**, 81–91 (2017)
5. Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., & Smola, A. J. A Kernel Statistical Test of Independence. In Advances in Neural Information Processing Systems, **20**, 585–592. (2008)
6. Guyon, I.: Results and analysis of the 2013 ChaLearn cause-effect pair challenge. In Proceedings of NIPS 2013 Workshop on Causality: Large-scale Experiment Design and Inference of Causal Mechanisms (2014)
7. Husson F., Lê S., Pagès J. Analyse de données avec R, Rennes : Presses Universitaires de Rennes (2009).

8. Kline R. B. Principles and Practice of Structural Equation Modeling. (3rd ed.) New-York : The Guilford Press. (2011)
9. Kummerfeld, E., Ramsey, J.: Causal Clustering for 1-Factor Measurement Models. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1655-1664. ACM, 2016.
10. Le S., Josse J., Husson F.. FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software, **25**(1), 1-18. (2008). 10.18637/jss.v025.i01
11. MacCallum, R. C., & Austin, J. T.: Applications of Structural Equation Modeling in Psychological Research. Annual Review of Psychology, 51, 201–226. (2000)
12. Peters, J., Janzing, D., & Schölkopf, B.: Elements of causal inference - Foundations and learning algorithms. Cambridge, Massachusetts: MIT Press. (2018)
13. Pfister N. and Peters J. dHSIC: Independence Testing via Hilbert Schmidt Independence Criterion. R package version 2.1. (2019). https://CRAN.R-project.org/package=dHSIC
14. R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
15. Rizzo M. and Szekely G. energy: E-Statistics: Multivariate Inference via the Energy of Data. R package version 1.7-7. (2019). https://CRAN.R-project.org/package=energy
16. Sen, A., & Sen, B. Testing independence and goodness-of-fit in linear models. Biometrika, **101**, 927–942. (2014).
17. Shimizu S., Hoyer P. O., and Hyvärinen A.. Estimation of linear non-Gaussian acyclic models for latent factors. Neurocomputing, 72, 2024-2027. (2009)
18. Silva R., Scheine R., Glymour C., and Spirtes P.: Learning the Structure of Linear Latent Variable Models. Journal of Machine Learning Research, 7, 191-246. (2006).
19. Spirtes, P., Zhang, K.: Causal discovery and inference: concepts and recent methodological advances. Applied Informatics, **3**(3), 1–28 (2016)
20. Szekely, G. J., Rizzo, M. L., & Bakirov, N. K.: Measuring and testing dependence by correlation of distances. The Annals of Statistics, **35**(6), 2769–2794 (2007)
21. Thoemmes, F.: The assumptions of direction dependence analysis. Multivariate Behavioral Research, 1-7. (2019)
22. Wiedermann W. & Li X.: Direction dependence analysis: A framework to test the direction of effects in linear models with an implementation in SPSS. Behavior Research Methods, **50**, 1581–1601 (2018)
23. Wiedermann, W., Merkle, E. C., & von Eye, A.: Direction of dependence in measurement error models. British Journal of Mathematical and Statistical Psychology, **71**(1), 117–145. (2018)
24. Wiedermann, W., Sebastian, J.: Direction Dependence Analysis in the Presence of Confounders: Applications to Linear Mediation Models Using Observational Data. Multivariate Behavioral Research, (2019a)
25. Wiedermann, W., Sebastian, J.: Sensitivity Analysis and Extensions of Testing the Causal Direction of Dependence: A Rejoinder to Thoemmes (2019), Multivariate Behavioral Research. (2019b)

Solving Hofstadter's Analogies using Structural Information Theory

Geerten Rijsdijk[1] and Giovanni Sileno[1]

[1]Informatics Institute, University of Amsterdam, Amsterdam
geerten.rijsdijk@student.uva.nl, g.sileno@uva.nl

Abstract. Analogies are common part of human life; our ability to handle them is critical in problem solving, humor, metaphors and argumentation. This paper introduces a method to solve string-based (symbolic) analogies based on hybrid inferential process integrating Structural Information Theory—a framework used to predict phenomena of perceptual organization—with some metric-based processing. Results are discussed against two empirical experiments, one of which conducted along this work, together with the development of a Python version of the SIT encoding algorithm PISA.

Keywords: Analogical reasoning · Symbolic Analogies · Compression · Structural Information Theory · Complexity

1 Introduction

Analogies are common part of human life; our ability to handle them is critical in problem solving, humor, metaphors and argumentation [8]. In psychology, analogy is seen as the process of understanding new information by means of structural similarities with previously acquired information [10], and analogical reasoning is one of the predominantly measured abilities on IQ tests. Because of their importance to cognition, analogies have interested researchers in the field of artificial intelligence. Systems for the computation of analogies have been created since the '60s for many different purposes such as solving puzzles based on objects in images [3], obtaining information by inference [9], understanding the development in analogical reasoning in children [16], or even as support in suggesting specialized care for patients with dementia [21]. Recent contributions in natural language processing [14] have suggested that analogical inference can be directly performed as vector operations on word vectors (e.g. `Paris` $\approx$ `France` + `Berlin` − `Germany`). However, even if some machine learning methods have proven to be unexpectedly good at reproducing some of these inferences, the overall results are not yet conclusive [18]. The centrality of analogies in human reasoning motivates to continue the effort to find a better understanding of their underlying mechanisms.

The aim of this paper is to offer an alternative solution to string-based (or symbolic) analogies as those proposed by Hofstadter [8]. The contribution is a hybrid inferential process integrating *Structural Information Theory* (SIT),

introduced to predict phenomena of perceptual organization, with some metric-based processing depending on the atomic components of the input. Thanks to an anonymous reviewer we discovered that such an application of SIT has been explored before [1], with a investigation on the algebraic properties of SIT extended with domain-dependent operators (e.g. *succ* to produce consecutive symbols). However, that work was presented before the creation of the minimal encoding algorithm PISA used in the present research to conduct our experiments. Even if preliminary, the results of the method we propose go beyond the state of art both in terms of the types of analogy it can deal with and its speed. Additionally, we report on the development of a Python version of the SIT encoding algorithm PISA, modified to consider various methods to compute (descriptive) complexity. The code used for this work is publicly available.[1]

The paper proceeds as follows: in the remainder of this section, Hofstadter's analogies and Structural Information Theory are briefly introduced. Section 2 outlines the analogy solving algorithm, and presents in detail the different components. Section 3 briefly discusses the Python implementation of PISA. In section 4, the algorithm is evaluated on two datasets, and its performance is compared to that of *Metacat*. The paper ends with a discussion and a conclusion.

1.1 Hofstadter's Analogies

Schematically, an analogy can often be expressed as "A is to B what C is to D" (also known as *proportional analogy*). In order to model and perform simple but relevant experiments on analogical reasoning, Douglas Hofstadter proposed a micro-world for analogy-making at the end of the '80s [15]. In this microworld, the objects used for the analogies are strings of letters. An example of such an analogy is:

$$ABC{:}ABD{::}BCD{:}?$$

which should be read as: "ABC is to ABD like BCD is to ?". The answer commonly given by respondents to this test is BCE.

In order to predict human answers, Hofstadter created a computer program called Copycat [15]. To complete a given analogy, the program works with "agents", which gradually build up structures representing the understanding of the problem, eventually reaching a solution. Later the Copycat program was improved to Metacat [13], which adds a memory, allowing the program to prevent itself from performing actions it has previously tried. *Metacat*, which was last updated in 2016[2], represents plausibly the state of the art of algorithms available for this problem.

1.2 Structural Information Theory

Structural Information Theory, or SIT, is a theory about perception with roots in *Gestalt* psychology. Central to SIT is the *simplicity principle*, in practice a

[1] http://github.com/GeertenRijsdijk/SIT_analogies
[2] http://science.slc.edu/~jmarshall/metacat/

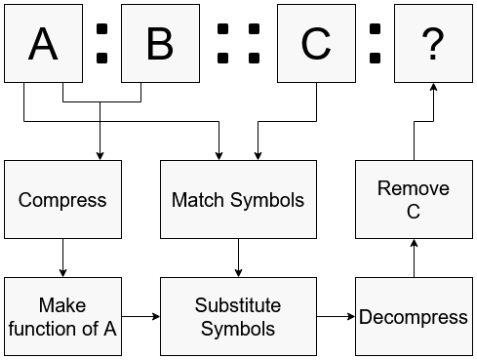


Fig. 1. Outline of the process used to answer an analogy of the form $A{:}B{::}C{:}?$.

formalization of Occam's razor: the simplest explanation for data is likely to be the correct one [11]. SIT has been empirically validated in several cognitive experiments with human participants [11,6].

SIT proposes to map the application of the simplicity principle to the Minimal Encoding Problem using the SIT language: given a string, use regularities to find an encoding with as little *complexity* as possible [11]. Such encoding can also be seen as a *compression*, since it may greatly decrease the amount of memory needed to store strings. There are three regularities/operators considered in the SIT coding language: *iteration* (I-form), *symmetry* (S-form) and *alternation* (A-form). They are defined in the following way:

I-form: $n * (\bar{y}) \Rightarrow yyy...y$ (n times, n ≥ 2)
S-form: $S[(\bar{x_0})(\bar{x_1})...(\bar{x_n}),(\bar{p})] \Rightarrow x_0x_1...x_npx_n...x_1x_0$
S-form: $S[(\bar{x_0})(\bar{x_1})...(\bar{x_n})] \Rightarrow x_0x_1...x_nx_n...x_1x_0$
A-form: $<(\bar{y})>/<(\bar{x_0})(\bar{x_1})...(\bar{x_n})> \Rightarrow yx_0yx_1...yx_n$
A-form: $<(\bar{x_0})(\bar{x_1})...(\bar{x_n})>/<(\bar{y})> \Rightarrow x_0yx_1...yx_ny$

(The p in a S-form is an optional element, called *pivot*.) These regularities have been proven to be the only ones which are *holographic* and *transparent* [5]. Holographic means invariant under growth, e.g. a repetition of n symbols can have that same symbol added to it forever, and will remain a repetition. Transparent means that the arguments of a regularity occur linearly in the original string it encodes. With alternations, the arguments can occur non-consecutively, they still occur in the same linear order.

2 Analogy Solving

Analogies rely on a perceived underlying structures (see e.g. Structural Mapping Theory [4]). Both Copycat [15] and Metacat [13] are based on the idea that there are structures on one side of the analogy that need to be replicated on the other side. Their "agents" are functionally meant to identify the structures to

be mapped. At functional level, however, what these agents do is nothing else than *compressing* the symbolic input, and this means that other compressors may work as well.

Figure 1 outlines the higher-level process that was followed to produce an analogical inference. In an analogy of form A:B::C:?, it is expected that there exists a certain structure in the left-hand side A:B, which can be extracted by means of some compression method. By applying this same structure to the partially available right-hand side C, decompressing the resulting code and taking away the part C, a possible answer to the analogy can be found.

2.1 Structural compression

The SIT encoding defines a way—empirically validated on perceptual experiments—to compress strings and is therefore a plausible candidate for describing regularities emerging from the input. Indeed, handling strings—here seen as ordered lists of characters, not as words from some language—seems to be based primarily on perceptual mechanisms rather than on semantics.

Applying SIT on analogical inference, we decided to extract the structure of A:B focusing on the concatenated string A+B. The idea of compressing A and B together, rather than separately, is inspired by the technique used in the famous paper by Li and Vitanji on the *Similarity Metric* [12], in which concatenations a.o. of DNA-sequences of two species were compressed to find a measure of their similarity. In our case, we used for compression the PISA algorithm (Parameter load plus ISA-rules) [7], a minimal coding algorithm proposed specifically for SIT [7]. For example, in the problem $ABC{:}ABD{::}IJK{:}?$, the minimal code (or compression) of the concatenated left-hand side $ABC{:}ABD$ would be $S[(AB),(C)]D$ or $<(AB)><(C)(D)>$. PISA is currently the most optimized algorithm for performing this task, being only weakly exponential. Along with this work, a Python implementation of this algorithm is presented.

2.2 Generating symbols from symbols

To apply the structure extracted from the left-hand side of the analogy (namely from A+B) to the right-hand side (C+?), this structure needs to be defined only as a function of symbols in A, as only the first part of the right hand is known. The symbols in B need to be generated from the symbols in A, and for this some invertible function capturing an adequate relationship between the atomic symbols is needed. The SIT coding language cannot do this; the only relationship this language considers is the *identity* relationship: denoting an atomic component of perception with a or z is completely arbitrary; the only assumption is that all a or z map to the same type of atomic component.

In contrast, Hofstadter's analogies seem to rely on some metrical information. For instance, d is the most natural answer to the analogy $a{:}b{::}c{:}?$, obtained by *contrasting* the b and a (as objects in an alphabet) and applying the output of this operation on c. The role of contrast in concept construction, similarity and description generation is indeed central [2,20]. For alphabetic characters as

in Hofstadter's analogies, contrast between symbols can be simply defined as a directional distance between the positions of those symbols in the alphabet. So, for example, the distance between b and a is 1, and the distance between a and e is -4. This allow us to rewrite symbols in B as one of the symbol in A and a directional distance. A challenge that arises with this approach is *which symbol the distance should be calculated from*. We attempted multiple approaches to this problem, some of which were more useful in different situations than others.

Previous symbol strategy The simplest way of deciding which symbol to calculate distance from is to choose the *previous symbol*. For example, in the analogy $a{:}bcd{::}i{:}?$, the left part $abcd$ can be described as $a(\$+1)(\$+1)(\$+1)$, where \$ refers here to the last symbol used in the code. This same structure can be applied to obtain the plausible (see section 4) right-hand side $i{:}jkl$.

Last new symbol strategy There are however analogies where this approach fails. Take the analogy $aba{:}aca{::}ada{:}?$, in which $abaaca$ gets encoded as $S[(a),(b)]$ $S[(a),(c)]$. When distances are applied, the code becomes $S[(a),(b)]$ $S[(a),((\$+2))]$. Now, when the symbol b is substituted by the symbol d, the decompressed code becomes $adaaca$, resulting in solution aca, whereas aea is a much more plausible answer. An approach solving this issue is choosing the *last new symbol* in the organization extracted using SIT. In the previous example, the last new symbol is b, the mapping would then result in $S[(a),(b)]$ $S[(a),((\$\$+1))]$, where \$\$ is the last new symbol used in the organization. Substituting b with d would result in $S[(a),(d)]$ $S[(a),((\$\$+1))]$, decompressing into the expected $adaaea$.

Same position strategy These approaches do not use the actual position of symbols in the input string, but this does seem to play a role at times. Consider for instance $ae{:}bd{::}cc{:}?$. Here, a plausible answer would be db, where the change applied to the string is an increase by 1 for the first element, and a decrease by 1 for the second. In a case like this, a position-based approach would be useful, resulting in code $ae(a+1)(e-1)$ or $ae(\$\$\$+1)(\$\$\$-1)$, where \$\$\$ is the object in A in the same position of the object in B. However, this approach cannot be used when parts A, B and C of the analogy have different lengths.

2.3 Symbol Substitution

Once the compression of A+B has been defined using only symbols present in A, a substitution (or replacement) of the symbols in A with the symbols in C can be performed. In order to do so, A and C need to be represented in ways that allow the mapping of their components. Here too, different strategies have been tried, selecting the best depending on the case.

Representation as Strings The simplest way to represent A and C is in their input form: strings, i.e. ordered lists of characters, in which each symbol counts as one element. However, there are cases in which this method is not applicable, as for instance when A and C do not share the same length.

Representation by Compression The number of elements in the representation can be reduced by compressing the string and seeing it as a list of symbols and *highest level operators*. For example, $ijjkkk$ can be compressed into $i2*(j)3*(k)$, which is then split into i, $2*(j)$ and $3*(k)$. When a highest level operator has been used to replace a symbol, this operator itself will count as one symbol for the purposes of calculating new symbols from distances. If a distance was calculated from a symbol that has since been replaced by an operator, the entire operator will be carried over as the new element, and each individual symbol in this operator is increased by the distance. Consider the analogy *abc*:*abd*::$ijjkkk$:?:

- structure of A+B: $<(ab)>/<(c)((\$+1))>$
- structure of C: $i\ 2*(j)\ 3*(k)$
- substutition: $(a{\rightarrow}i), (b{\rightarrow}2*(j)), (c{\rightarrow}3*(k))$
- structure of C+D: $<(i2*(j))>/<(3*(k))((\$+1))>$
- distances removed: $<(i2*(j))>/<(3*(k))(3*(l))>$
- decompression: $ijjkkkijjlll$
- result D: $ijjlll$

Representation by Chunking Alternatively, A and C can be represented in terms of *chunkings*, meaning divisions of the concatenated input symbol strings into chunks. These chunks can then be replaced as if they concerned one element. For example, *abcd* could be chunked into $[ab, cd]$, $[a, bc, d]$, $[a, b, c, d]$, and so on. For A, the chunkings are derived from the compression of A+B, e.g.:

- $[a, b, c]$ is a chunking of *abc* in the compression $S[(a)(b), (c)]c$.
- $[ab, c]$ is a chunking of *abc* in the compression $ab2*(c)(\$+1)(\$+1)$.
- $[ab, c]$ is a chunking of *abc* in the compression $<(ab)>/<(c)((\$+1))>$.
- $[abc]$ is a chunking of *abc* in the compression $2*(abc)$.

For C, there is no pre-existing structure present that determines how it should be chunked. However, we considered sound to chunk C in such a way that it best corresponds to the chunking of A.

We call the process of creating a chunking of C as similar as possible to a chunking of A *chunking-element matching*, and works as follows: a list is created for the number of symbols in each element in the chunking of A. (e.g. $[a, bc, def]$ results in $[1, 2, 3]$). Next, if the sum of the numbers does not equal the number of symbols in C, as a simple heuristic, the largest number in this list is increased/decreased. Now, this list of numbers can be used to split C into chunks, which can be used to substitute the original elements of the chunking of A. The following example shows how this chunking matching is used to solve an analogy *abc*:*abd*::$ijklm$:?

- structure of A+B: $<(ab)>/<(c)((\$+1))>$
- chunking of A in A+B: $[ab, c]$
- lengths of chunking elements of A: $[2, 1]$
- matching chunking lengths of C: $[4, 1]$
- chunking of C: $[ijkl, m]$

- substitution: $(ab \rightarrow ijkl), (c \rightarrow m)$
- structure of C+D: $<(ijkl)>/<(m)((\$ + 1))>$
- distances removed: $<(ijkl)>/<(m)(n)>$
- decompression: $ijklmijkln$
- result D: $ijkln$

Representation by Consecutive Chunking A special type of chunking is a *consecutive chunking*. A chunking is consecutive if all elements of the chunking contain exactly one symbol; the elements of the chunking together form all arguments of a single operator. For instance, $[a, b, c]$ is a consecutive chunking of *abc* in the code $<(a)(b)(c)>/<(d)>$, and $[a, b, c]$ is a consecutive chunking of *abc* in the code $S[(a)(b), (c)]$. Consecutive chunkings can be substituted in a special way: instead of replacing individual symbols or elements, the entire chunking can be replaced by a new chunking. The new chunking has one element for every symbol in C. When this consecutive chunking is applied, the entire argument string forming A is replaced with this new chunking (with the corresponding number of parentheses per element). For instance in *abc*:*cba*::$ijklm$:?:

- structure of A+B: $S[(a)(b)(c)]$
- consecutive chunking of A in A+B: $[a, b, c]$
- consecutive chunking of C: $[i, j, k, l, m]$
- substitution: $([(a), (b), (c)] \rightarrow [(i), (j), (k), (l), (m)])$
- structure of C+D: $S[(i)(j)(k)(l)(m)]$
- decompression: $ijklmmlkji$
- result D: $mlkji$

2.4 Structure in Parameters

Consider the analogy *aaabb*:*aabbb*::*eeeeef*:?. One way to look at it is that the analogy simply swaps the number of times the first symbol occurs with the number of times the second symbol occurs. However, PISA assigns to A+B the structure $3 * (a)S[2 * ((b))(a)]b$, which does not seem to capture this intuition, as there is no symbol substitution that results in the expected answer *efffff*. The core of this analogy problem does not lie in the structure of the symbols, but in the structure of the structure. By looking at the structure as a series of iterations, this becomes clear; in $3 * (a)2 * (b)2 * (a)3 * (b)$, the parameters of the iterations form a symmetry, namely $S[(3)(2)]$. It is this symmetry that forms a plausible basis for solving this analogy. The function written to apply this strategy is separated from the rest of the algorithm. It encodes A+B as a sequence of iterations. Next, the parameters of these iterations are compressed, distances are added and symbol substitution is performed in essentially the same way as described before. This results in new parameters which, combined with symbol substitution on the actual symbols, can produce an answer to the analogy.

However, problems of this type can easily become more complex. When the parameters of a structure can have a structure, the parameters of *that* structure could also have a structure, which could again have parameters with some structure. With larger codes, this 'parameter depth' could become very high.

Algorithm 1: PISA-based compressor

```
Function compress_pisa(graph)
    new_hyperstrings = []
    for hyperstring h in graph do
        Q = QUIS(h)
        Create and encode S-graphs of h using Q
        for w in h.nodes[1 ... N] do
            Create and encode A-graphs of h using Q, up to node w
            for v in h.nodes[w ... 0] do
                find best possible code for v→w
                add best code to h as edge v→w
                for u in h.nodes[0 ... v] do
                    if c(u, w) > c(u,v) + c(v,w) then
                        new_code = u→v + v→w
                        add new_code to h as edge u→w
        add h to new_hyperstrings;
    return combine(new_hyperstrings);
```

Furthermore, relationships between parameters at different 'depths' are also possible. Take for instance the analogy *abc*:*aaabbbccc*::*abcd*:. The structure of the parameters could be written as $3*(1)3*(3)$ (neglecting the internal relationships between symbols). To get to a plausible answer *aaaabbbbccccdddd*, there would need to be a relationship between the two 3s out of brackets, and the 3 inside the brackets, which are at different parameter depths. In short, structure in parameters can be very complex. In this work, it has only been explored at a surface level. Other configurations are left to future work.

2.5 Inversion trick

When A:B::C:? does not have a structure that is easily worked with, it is also possible to rearrange the analogy to hopefully obtain better answers. This rearrangement is analogous to this numeric equivalence: $\frac{a}{b}=\frac{c}{d} \Leftrightarrow \frac{a}{c}=\frac{b}{d}$. In our case, while the two forms of analogies might often result in the same answers, one of the two forms might be more solvable using the approach proposed here. An example of this is the analogy *abac*:*adae*::*baca*:?. A+B has a structure of form $<(a)>/<(.)>$, while the left side has a structure of form $<(.)>/<(a)>$. This change in structure is a problem for our algorithm, since it tries to apply the same structure to C+D. Changing the analogy to *abac*:*baca*::*adae*:? results in the structure $S[(a)(bac)]$, which is a structure the solver can deal with more easily.

3 Python implementation of PISA

A string of length N can be represented by up to a superexponential $O(2^{Nlog(N)})$ number of codes [7]. To find the one with the lowest complexity, one could

generate each possible code and compare all of these. For long strings this can be very time consuming. The PISA (Parameter load plus ISA-rules) algorithm was designed to efficiently find the minimal coding of a string in the SIT coding language [7]. PISA is significantly faster than a method which generates all possible encodings, being only weakly exponential. A dissection of the PISA algorithm can be found in chapter 5 of *Simplicity in Vision* [6]. Here, we will briefly discuss our re-implementation in Python.

The PISA-based compressor created for this research is written in an object-oriented fashion (the original version[3], being in C, does not support classes), with a central *hyperstring* class. The general outline of the algorithm written for this research can be seen in Algorithm 1. The algorithm processes each hyperstring in the input graph separately (line 3). In line 4, the QUIS algorithm [6] is called for each hyperstring to create an intermediate structure to more efficient representations. The output matrix is indeed used in line 5 and 7 to create S- and A-graphs, representing symmetries and alternations present in the hyperstring. These graphs are themselves also encoded using this function. Lines 6 and 8 loop over every combination of two nodes (v, w) in the hyperstring. For each pair, lines 9 looks for the best possible code for the substring between these two nodes. Using the complexity metric, this best code is chosen from: the current code; the best possible iteration, if any; the best possible symmetry, if any, calculated using the S-graph that has a pivot halfway in between the two nodes; the best possible alternation, if any, calculated for every A-graph. Once the best code has been selected, line 10 adds an edge representing the code to the hyperstring. Next, line 11 iterates over every node that comes before node v. Line 12 looks at the complexities of the codes between u, v and w. If the complexity of the edge $u \rightarrow w$ can be reduced by creating a combination of the codes in edges $u \rightarrow v$ and $v \rightarrow w$, this is done. At the end of the algorithm, all encoded hyperstrings are recombined into a new graph. This graph is then returned.

Given a hyperstring that represents a mere string, the same hyperstring is returned with added edges that represent the best code for each substring, and the code of the edge connecting the first and last node of the hyperstring will be the best encoding of the entire string.

3.1 Configurable Complexity Metric

Besides reproducibility, a major reason why we reimplemented PISA was to gain control over its components, in particular the way in which complexity is measured. The original PISA relies only on one metric, the I_{new} load [7]. We considered instead as basis the more general principles of Kolmogorov complexity [19]. At a more fundamental level, SIT has been conceived for structural information, but analogies require also to look at some metrical information. The complexity metric considered here calculates complexity by taking the number of symbols used for the code and adding, for each operator in the code, a certain value. This value might differ across operators, and can be adjusted later with

[3] https://ppw.kuleuven.be/apps/petervanderhelm/doc/pisa.html

empirical data to find the values for optimal performance of the analogy solving algorithm. Indeed, the I_{new} load does not allow for adjusting of parameters for aligning to human answers and seems somewhat unintuitive, assigning for example the same complexity to the codes a and $9 * (a)$.

3.2 Other differences with PISA

This implementation relies heavily on the theoretical concepts of hyperstrings, S-graphs and A-graphs as PISA does. However, in some points we found the exact working of PISA to be unclear, and that meant that we had to fill in the blanks. The following list outlines the major differences between the two algorithms.

- The explanation of PISA in [6] mentions 'updating its database of S- and A-graphs' at the end of the first for loop. It is however not clear how this update is done. In the proposed compressor the graphs are not updated, but recreated each time.
- PISA updates the A-graphs at the *end* of the first for loop, while this compressor recreates the A-graphs at the *start* of the first for loop. A small exception to this is present in the code; at the end of each v loop, the algorithm does update the repeats of right a-graphs with the encodings of the $v \rightarrow w$ edge. This is not necessary for left a-graphs due to the algorithm encoding the string left to right.
- PISA updates the S-graphs at the *end* of the first for loop, while this compressor creates the S-graphs before the first for loop.
- PISA always returns the one code with the lowest complexity, while this compressor returns a Graph object. In this object, the edge connecting the first and last nodes also represents the edge with the lowest complexity, but other paths represent other codes, which consist of optimally encoded substrings which together form the whole string. This enables us to consider sub-optimal codes as well.

4 Results

To evaluate the proposed analogy solving algorithm, the answers generated by it will be compared against two sets of human answers obtained in distinct experiments. Furthermore, the answers generated by the proposed solver will also be compared to the answers generated by *Metacat* [13].

4.1 Murena's dataset

Table 1 reports data published by Murena et. al. [17] on human answers for analogy tests. In their experiment, 68 participants were asked to solve analogies following the template ABC:ABD::X:?. The left-hand side of the analogy remained the same during the experiment, but the X changed in every test. For each X, the data shows the two most common answers given by participants,

Given X	Solutions	Selected by	P_s	P_M	Given X	Solutions	Selected by	P_s	P_M
IJK	IJL	**93%**	**1**	**1**	BCD	BCE	**81%**	2	**1**
	IJD	2.9%	-	-		BDE	5.9%	**1**	-
BCA	BCB	**49%**	3	2	IJJKKK	IJJLLL	**40%**	**1**	2
	BDA	43%	**1**	**1**		IJJKKL	25%	2	**1**
AABABC	AABABD	**74%**	**1**	**1**	XYZ	XYA	**85%**	**1**	-
	AACABD	12%	-	-		IJD	4.4%	-	-
IJKLM	IJKLN	**62%**	**1**	**1**	RSSTTT	RSSUUU	**41%**	**1**	**1**
	IJLLM	15%	-	-		RSSTTU	31%	2	-
KJI	KJJ	**37%**	**1**	**1**	MRRJJJ	MRRJJK	**28%**	2	**1**
	LJI	32%	-	2		MRRKKK	19%	**1**	2
ACE	ACF	**63%**	**1**	**1**					
	ACG	8.9%	-	-					

Table 1. Human answers to analogies of form ABC:ABD::X:? from the Murena dataset 1, along with at which position the same answers were given by the solving algorithm proposed in this project (P_s) and *Metacat* (P_M).

as well as the percentage of participants that chose that answer.[4] The last two columns in the Table 1 show the performances of the analogy solving algorithm proposed in this project (P_s) and *Metacat* (P_M) ins terms of the position in which that answer was generated (e.g. a 1 means it was the best answer, 2 means it was the second best, etc.). A dash indicates that the answer was not generated at all. As for speed, the lack of a built-in way to measure the time *Metacat* uses for compression made it difficult to perform an empirical speed comparison. However, when working with *Metacat*, it was clear that this method is much slower than the solver implemented here, sometimes taking more than 10 minutes to generate a single answer. For this reason, only two answers per question were generated using *Metacat*.

The table shows that, for this dataset, the most common answer to the problem is always generated by our solver. Furthermore, the top answer generated by the solver is always one of the two most common answers by the participants. Both of these observations are also true for *Metacat*, with the exception of the problem XYZ, for which it produced none of the answers given by participants. However, there are also answers given by human participants that the solvers did not generate. Overall, the most common human answer matched the top answer 8/11 times (72.7%), both for our solver and *Metacat*. The most common participant answer was in the top 2 generated answers 10/11 times (90.9%) for both algorithms. Answers given by participants were generated 16/22 times (72.7%) for our solver and 14/22 times (63.7%) for *Metacat*.

4.2 Our Dataset

The Murena testset is quite small and the analogies it presents follow all the same template. For this reason, a second testset was constructed on purpose for this work, consisting of 20 more complex analogies. 35 participants (18 male,

[4] In the original experiments some questions were repeated to see the influence of having previously faced similar problems. Since the solving algorithm in this project runs independently of previous answers, repeated questions were omitted here.

Given problem	Solutions	Selected by	P_s	P_M
ABA:ACA::ADA:?	AEA	**97.1%**	**1**	**1**
	AFA	2.9%	-	-
ABAC:ADAE::BACA:?	DAEA	**60%**	2	-
	BCCC	28.6%	21	-
AE:BD::CC:?	DB	**68.5%**	3	**1**
	CC	17.1%	-	2
ABBB:AAAB::IIIJJ:?	IIJJJ	**57.1%**	**1**	-
	JJIII	14,3%	-	-
ABC:CBA::MLKJI:?	IJKLM	**88.6%**	**1**	**1**
	-	-	-	-
ABCB:ABCB::Q:?	Q	**100.0%**	**1**	-
	-	-	-	-
ABC:BAC::IJKL:?	JIKL	**54.3%**	-	-
	KIJL	14.3%	2	-
ABACA:BC::BACAD:?	AA	**57.1%**	**1**	-
	BCD	31.4%	-	-
AB:ABC::IJKL:?	IJKLM	**85.7%**	**1**	**1**
	IJKLMN	11.4%	-	-
ABC:ABBACCC::FED:?	FEEFDDD	**91.4%**	2	1
	-	-	-	-
ABC:BBC::IKM:?	JKM	**57.1%**	7	-
	KKM	37.1%	2	-
ABAC:ACAB::DEFG:?	DGFE	**68.6%**	2	-
	FGDE	14.3%	**1**	-
ABC:ABD::CBA:?	DBA	**51.4%**	**1**	2
	CBB	45.7%	2	**1**
ABAC:ADAE::FBFC:?	FDFE	**94.3%**	**1**	-
	FDFA	2.9%	-	-
ABCD:CDAB::IJKLMN:?	LMNIJK	**80.0%**	-	-
	-	-	-	-
ABC:AAABBBCCC::ABCD:?	AAABBBCCCDDD	**74.3%**	**1**	**1**
	AAAABBBBCCCCDDDD	17.1%	-	-
ABC:ABBCCC::ABCD:?	ABBCCCDDDD	**85.7%**	-	-
	ABBCCCDDD	8.6%	**1**	-
ABBCCC:DDDEEF::AAABBC:?	DEEFFF	**77.1%**	**1**	-
	DCCDDF	8.6%	-	-
A:AA::AAA:?	AAAAAA	**62.8%**	**1**	-
	AAAA	25.7%	2	**1**
ABBA:BAAB::IJKL:?	JILK	**71.4%**	-	-
	JIJM	11.4%	5	-

Table 2. Human answers to analogies collected in this project experiment, along with at which position the same answers were given by the solving algorithm proposed in this project and *Metacat*.

17 female, average age 26.8) were asked to solve the analogies in the testset, the results of which can be seen in Table 2. As before, the table shows the top two answers given by participants, as well as the percentage of participants that gave each answer. In some cases, only the top answer is given. This is done when either all participants gave the same answer, or when there were multiple answers tied for second which all had only one participant.

In this testset, the most common answer given by participants was generated by the solver 16/20 times (80%), whereas it was generated by *Metacat* only 7/20 times (35%). The top answer given by the solver was in the top two participant answers 13/20 times (65%), whereas the top answer generated by *Metacat* was in

the top two participant answers 8/20 times (40%). The most common participant answer matched the top generated 10/20 times (50%) for the solver, and 6/20 times (30%) for *Metacat*.

4.3 Complexity values

The complexity values or weights of iteration, symmetry and alternation operators were chosen to optimize the results of the solver on the two testsets. These variables were tested with values ranging from 0.8 to 1.2, with steps of 0.1. Each possible combination of those values was tested on how highly they ranked the participant-given answers amongst all answers. Overall, it was found that small differences in the values often did not change much about the rankings, suggesting that there was not really a risk of overfitting. On a larger scale, the following requirements seem to yield the best results: the weight of iterations should be less than 1; of symmetries less than 1; of alternation more than 1. The final weights chosen were 0.85 for iterations, 0.9 for symmetries and 1.1 for alternation.

5 Discussion

The goal of this project was to use SIT compression as the basis for an analogy solving algorithm. The analogy solving algorithm has shown promising results on the test set created by Murena et. al [17]. However, this test set is very limited, having only 11 questions, each following the same template. The lack of variety and complexity motivated the creation of a second testset.

When compared to *Metacat*, our solving algorithm has shown to achieve similar results on the first testset. This is likely because the questions in this testset share a similar structure which both solvers seem to be able to deal with. On the second testset, our solving algorithm achieves drastically better results. It should, however, be noted that *Metacat* uses randomness in its procedure to generate answers. Therefore, different runs of the algorithm on the problem could result in different, and possibly better, answers. Furthermore, for this comparison, only the first two answers generated by *Metacat* were used, but *Metacat* can often generate more answers than that. The choice to only consider the top two answers was made due to time constraints: generating a single answer using *Metacat* can, in some cases, take up to 10 minutes, against a few seconds for our solver.

It is important also to note that problems in the second testset were created after the implementation of the solving algorithm, and with the capabilities of this solver in mind. Because of this, for many of the questions it was predictable beforehand whether the solver would produce intuitive answers. Therefore, the percentages of correct answers should not weigh heavily in the evaluation of the algorithm. Instead, the set should be used as a showcase of what types of analogies the algorithm can and cannot deal with.

Answers of the solver are ranked by the complexities of the codes they originate from, e.g. answers such as $BCCC$ (28.6%) to $ABAC{:}ADAE{::}BACA{:}$

?, DB (68.5%) to $AE{:}BD{::}CC{:}?$, JKM(57.1%) to $ABC{:}BBC{::}IKM{:}?$ and $JIJM$(11.4%) to $ABBA{:}BAAB{::}IJKL{:}?$. Despite these answers being chosen by (fairly) significant percentages of the participants, they do not rank highly amongst the answers generated by the solver. This results hints that there exist better strategies not adequately taken into account. The reasoning behind these answers (most likely) relies on applying positional distances in the left-hand side of the analogy to the right-hand side. The most significant case of this is the answer $BCCC$ to $ABAC{:}ADAE{::}BACA{:}?$, which the solver ranks lower than 20 other solutions, despite being picked by over a quarter of the participants. Future work on this project could look at alternate ways which could, either in combination with complexity or on their own, rank answers generated by the solver in a way that corresponds better to human answers.

The answer $LMNIJK$ (80.0%) to $ABCD{:}CDAB{::}IJKLMN{:}?$ might tell us something the cognitive equivalent of what in this project is called chunking (section 2.3). It seems that the structure that best corresponds to participants' interpretation of this problem is $S[(ab)(cd)]$, which essentially represents a swapping of ab and cd in part A to get part B. The same structure in the right-hand side of the analogy that corresponds to the top answer is $S[(ijk)(lmn)]$. This way of symbol substitution corresponds to *chunking element matching*, described in section 2.3; whereas the method used in this project tries to keep as many elements of the chunking as possible at the same length (which results in structures like $S[(ij)(klmn)]$ or $S[(ijkl)(mn)]$), the 3-3 division suggests a preference to maintain the same ratio between the chunking elements.

Finally, other answers that cannot be solved by the algorithm are the ones discussed in section 2.4, where there are relationships between iteration parameters at different levels. In the test, such relationships are (most likely) used for answer $AAAABBBBCCCCDDDD$ (17.1%) to problem $ABC{:}AAABBBCCC{::}ABCD{:}?$, and answer $ABBCCCDDDD$ (85.7%) to problem $ABC{:}ABBCCC{::}ABCD{:}?$. These answers suggest that such relationships are indeed understood and used by participants, although this begs the question of how complex these relationships can be before participants will no longer base their answer on them.

6 Future developments

Structural Information Theory has shown itself to be a useful tool for analogy solving, although it cannot do this on its own. The lack of metrical information, or a way to define relationships between symbols, resulted in the need for a way of defining symbols as distances from other symbols, as well as a way of choosing which symbol to calculate from. Similarly, the necessity to apply structure from one part of an analogy to another entailed the need for a method for symbol substitution. In this work we introduced with some success different intuitive heuristics/strategies for these two aspects, but a general, unifying theory is needed. Additionally, test data confirms that, sometimes, the structure of the symbols has structure itself (section 2.4). A unifying theory based on Kolmogorov complexity might predict that further depth is considered only if yields a reduction of complexity, and this is a required focus for future works.

References

1. Dastani, M., Indurkhya, B., Scha, R.: Analogical projection in pattern perception. Journal of Experimental and Theoretical Artificial Intelligence **15**(4), 489–511 (2003)
2. Dessalles, J.L.: From Conceptual Spaces to Predicates. Applications of conceptual spaces: The case for geometric knowledge representation pp. 17–31 (2015)
3. Evans, T.G.: A program for the solution of a class of geometric-analogy intelligence-test questions. Tech. rep., Air Force Cambridge Research Labs (1964)
4. Gentner, D.: Structure-mapping: A theoretical framework for analogy. Cognitive Science **7**(2), 155–170 (1983)
5. A van der Helm, P., J van Lier, R., LJ Leeuwenberg, E.: Serial pattern complexity: Irregularity and hierarchy. Perception **21**(4), 517–544 (1992)
6. Van der Helm, P.A.: Simplicity in vision: A multidisciplinary account of perceptual organization. Cambridge University Press (2014)
7. van der Helm, P.A.: Transparallel mind: Classical computing with quantum power. Artificial Intelligence Review **44**(3), 341–363 (2015)
8. Hofstadter, D.R.: Analogy as the core of cognition. The analogical mind: Perspectives from cognitive science pp. 499–538 (2001)
9. Hummel, J.E., Holyoak, K.J.: Lisa: A computational model of analogical inference and schema induction. In: Proceedings of the Cognitive Science Society. pp. 352–357. Lawrence Erlbaum Associates Hillsdale, NJ (1996)
10. Itkonen, E.: Analogy as structure and process: Approaches in linguistics, cognitive psychology and philosophy of science, vol. 14. John Benjamins Publishing (2005)
11. Leeuwenberg, E., Van der Helm, P.A.: Structural information theory: The simplicity of visual form. Cambridge University Press (2013)
12. Li, M., Chen, X., Li, X., Ma, B., Vitányi, P.M.: The similarity metric. IEEE transactions on Information Theory **50**(12), 3250–3264 (2004)
13. Marshall, J.B.: Metacat: A self-watching cognitive architecture for analogy-making. In: Proceedings of the Cognitive Science Society. vol. 24 (2002)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Advances in Neural Information Pro-cessing Systems 26 (NIPS 2013). pp. 3111–3119 (2013)
15. Mitchell, M., Hofstadter, D.R.: The emergence of understanding in a computer model of concepts and analogy-making. Physica D: Nonlinear Phenomena **42**(1-3), 322–334 (1990)
16. Morrison, R.G., Doumas, L.A., Richland, L.E.: A computational account of children's analogical reasoning: balancing inhibitory control in working memory and relational representation. Developmental science **14**(3), 516–529 (2011)
17. Murena, P.A., Dessalles, J.L., Cornuéjols, A.: A complexity based approach for solving hofstadter's analogies. In: ICCBR (Workshops). pp. 53–62 (2017)
18. Rogers, A., Drozd, A., Li, B.: The (too many) problems of analogical reasoning with word vectors. *SEM 2017 - 6th Joint Conference on Lexical and Computational Semantics, Proceedings pp. 135–148 (2017)
19. Shen, A., Uspensky, V.A., Vereshchagin, N.: Kolmogorov complexity and algorithmic randomness, vol. 220. American Mathematical Soc. (2017)
20. Sileno, G., Bloch, I., Atif, J., Dessalles, J.L.: Similarity and Contrast on Conceptual Spaces for Pertinent Description Generation, vol. 10505 LNAI (2017)
21. Zachos, K., Maiden, N., Pitts, K., Jones, S., Turner, I., Rose, M., Pudney, K., MacManus, J.: Digital creativity in dementia care support. International Journal of Creative Computing **1**(1), 35–56 (2013)

A Semantic Tableau Method for Argument Construction

Nico Roos

Data Science and Knowledge Engineering, Maastricht University
roos@maastrichtuniversity.nl
https://dke.maastrichtuniversity.nl/nico.roos/

Abstract. A semantic tableau method, called an argumentation tableau, that enables the derivation of arguments, is proposed. First, the derivation of arguments for standard propositional and predicate logic is addressed. Next, an extension that enables reasoning with defeasible rules is presented. Finally, reasoning by cases using an argumentation tableau is discussed.

1 Introduction

The semantic tableau method is used for (automated) reasoning with different logics such as the standard propositional and predicate logic [5], several modal logics, description logics, etc. Although a semantic tableau proof can be viewed as an argument for a claim / conclusion, it is not similar to arguments studied in argumentation systems; see for instance: [3, 4, 11–15, 18, 19, 21, 25]. This raises the question whether the semantic tableau method can be used to derive proper arguments for claims / conclusions.

We will address this question by first investigating a semantic tableau method, called an argumentation tableau, for the derivation of arguments in standard propositional and predicate logic. The use of arguments becomes more interesting when dealing with defeasible information. We therefore will investigate next the derivation of arguments when propositional and predicate logic are extended with defeasible rules.

Reasoning by cases is a problem for many argumentation systems that use an underlying language that allows for disjunctive information. Moreover, approaches that support reasoning by cases, do not agree on how rebutting attacks should be handled within a case [2, 6, 12, 13, 19]. We also will investigate reasoning by cases using an argumentation tableau.

2 Preliminaries

This section presents the notion of an argument that will be used in the discussion of the argumentation tableau that is proposed in this paper.

We assume a standard logic such as propositional or predicate logic. The language of the logic will be denoted by $\mathcal{L}$. We also assume that the language $\mathcal{L}$

contains the symbols $\top$ denoting *true*, and $\bot$ denoting *false*. In case of predicate logic, the set of ground terms is denoted by $\mathcal{G}$.

Since this paper focuses on argumentation, we need a definition of an argument. Toulmin [23] views an argument as a support for some *claim*. The support is grounded in *data*, and the relation between the data and the claim is the *warrant*. Here, we use the following definition.

Definition 1. *A couple $A = (\mathcal{S}, \varphi)$ is called an argument where φ is a proposition said to be its conclusion, and $\mathcal{S}$ is a set said to be its support; its elements are called supporting elements. It is worthwhile observing here that this definition is very general and a many couples might be qualified as arguments.*

In case of propositional and predicate logic, the support $\mathcal{S}$ is a set of propositions from the language $\mathcal{L}$. Generally, $\mathcal{S}$ contains the set of premises used to derive the supported proposition φ. So, $\mathcal{S} \vdash \varphi$. In special applications, such as Model-Based Diagnosis, we may restrict $\mathcal{S}$ to assumptions about the normal behavior of components.

We may extend a standard logic with a set of defeasible rules. Defeasible rules are of the form:

$$\varphi \rightsquigarrow \psi$$

in case of propositional logic, and of the form:

$$\varphi(\mathbf{x}) \rightsquigarrow \psi(\mathbf{x})$$

in case of predicate logic. Here φ and ψ are propositions from the language $\mathcal{L}$, and $\mathbf{x}$ is a sequence of free variables. The free variables denote a set of ground instances of the defeasible rule $\varphi(\mathbf{x}) \rightsquigarrow \psi(\mathbf{x})$. We do not use the universal quantifier because the rule is not a proposition that belongs to the language $\mathcal{L}$. It is an additional statement about preferences that need not be valid for every ground instance.

We use $\Sigma \subseteq \mathcal{L}$ to denote the set of available information and we use D to denote the set of available rules. Moreover, we use $\overline{D} = \{\varphi(\mathbf{t}) \rightsquigarrow \psi(\mathbf{t}) \mid \varphi(\mathbf{x}) \rightsquigarrow \psi(\mathbf{x}) \in D, \mathbf{t} \in \mathcal{G}^n\}$ to denote the set of ground instances of the defeasible rules with n free variables in case of predicate logic, and $\overline{D} = D$ in case of propositional logic.

Defeasible rules are used in the construction of arguments. Whenever we have a support $\mathcal{S}'$ for the antecedent φ of a defeasible rule $\varphi \rightsquigarrow \psi$, we can create a supporting element $(\mathcal{S}', \varphi \rightsquigarrow \psi)$, which can be used to support ψ. The arguments that can be constructed are defined as:

Definition 2. *Let $\Sigma \subseteq \mathcal{L}$ be the initial information and let D be a set of defeasible rules. An argument $A = (\mathcal{S}, \psi)$ with premises $\bar{A}$, defeasible rules $\tilde{A}$, last defeasible rules $\vec{A}$, supported proposition (claim / conclusion) $\hat{A}$, and supporting propositions $\hat{\mathcal{S}}$ of $\hat{A}$, is defined as:*

- *If $\psi \in \Sigma$, then $A = (\{\psi\}, \psi)$ is an argument.*
  $\bar{A} = \{\psi\}$. $\tilde{A} = \varnothing$. $\vec{A} = \psi$. $\hat{\mathcal{S}} = \{\psi\}$.

- *If* $A_1 = (\mathcal{S}_1, \varphi_1), \ldots, A_k = (\mathcal{S}_k, \varphi_k)$ *are arguments and* $\{\varphi_1, \ldots, \varphi_k\} \vdash \psi$, *then* $A = (\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_k, \psi)$.
  $\bar{A} = \bar{A}_1 \cup \cdots \cup \bar{A}_k$. $\tilde{A} = \tilde{A}_1 \cup \cdots \cup \tilde{A}_k$. $\vec{A} = \vec{A}_1 \cup \cdots \cup \vec{A}_k$. $\hat{A} = \psi$. $\hat{\mathcal{S}} = \hat{\mathcal{S}}_1 \cup \cdots \cup \hat{\mathcal{S}}_k$.
- *If* $A' = (\mathcal{S}', \varphi)$ *is an argument and* $\varphi \rightsquigarrow \psi \in \overline{D}$ *is a defeasible rule, then* $A = (\{(\mathcal{S}', \varphi \rightsquigarrow \psi)\}, \psi)$ *is an argument.*
  $\bar{A} = \bar{A}'$. $\tilde{A} = \{\varphi \rightsquigarrow \psi\} \cup \tilde{A}'$. $\vec{A} = \{\varphi \rightsquigarrow \psi\}$. $\hat{A} = \psi$. $\hat{\mathcal{S}} = \{\psi\}$.

$A = (\mathcal{S}, \psi)$ *is a* minimal *argument iff (1)* $\mathcal{S}$ *is a minimal set such that* $\hat{\mathcal{S}} \vdash \psi$, *and (2) for every* $(\mathcal{S}', \alpha \rightsquigarrow \beta) \in \mathcal{S}$, $(\mathcal{S}', \alpha)$ *is a minimal argument.*

This abstract representation of arguments is based on the representation of arguments proposed in [18, 19]. Note that for every argument, there exists a corresponding minimal argument supporting the same conclusion.

We will use a graphical representation of an argument for human readability. The argument for an inconsistency:

$$A = (\{(\{(\{p \vee q, \neg q\}, p \rightsquigarrow r), (\{s\}, s \rightsquigarrow t)\}, r \wedge t \rightsquigarrow u), (\{v\}, v \rightsquigarrow w), \neg(u \wedge w)\}, \bot)$$

is graphically represented as:

$$A: \quad \begin{array}{rl} p \vee q \\ \neg q \end{array} \Big| \vdash p \rightsquigarrow r \Big| \quad \Big| \vdash r \wedge t \rightsquigarrow u \Big| \quad \Big| \vdash \bot$$
$$\qquad\qquad s \vdash s \rightsquigarrow t \Big| \qquad\qquad\qquad\

The undercutting arguments define an attack relation over the arguments. An undercutting argument $A = (\mathcal{S}, \mathbf{not}(\varphi \rightsquigarrow \psi))$ attacks every argument A' for which $\varphi \rightsquigarrow \psi \in \bar{A}'$ holds. Moreover, an undercutting argument $A = (\mathcal{S}, \mathbf{not}(\sigma))$ attacks every argument A' for which $\sigma \in \bar{A}'$ holds. In both cases, $A \longrightarrow A'$. We can use one the semantics for argumentation frameworks to determine the argument extensions; see for instance: [1, 7, 8, 10, 20, 24].

3 Basic Argumentation Tableau

A semantic tableau method is a proof system developed by Beth [5]. In the modern version of the method, the semantic tableau for propositional and predicate logic is a tree where each node represents a set of propositions. The set of propositions in a node of the tree is satisfiable if and only if the set of proposition in one of its child nodes is satisfiable. For convenience we will use Γ to denote a node of the semantic tableau as well as the set of propositions the node represents.

We are interested in arguments, which are propositions and their supports. Therefore we introduce an *argumentation tableau* of which each node Γ is a set of arguments.

Definition 4. *An argumentation tableau $\mathcal{T}$, is a tree of which each node Γ is of a set of arguments.*

The tableau rules of an argumentation tableau are similar to the rules of a traditional semantic tableau. The only difference is the supports for the propositions. The tableau rules for propositional logic arguments are:

$$\frac{(\mathcal{S}, \varphi \wedge \psi)}{(\mathcal{S}, \varphi), (\mathcal{S}, \psi)} \qquad \frac{(\mathcal{S}, \varphi \vee \psi)}{(\mathcal{S}, \varphi) \mid (\mathcal{S}, \psi)}$$

$$\frac{(\mathcal{S}, \varphi \rightarrow \psi)}{(\mathcal{S}, \neg\varphi) \mid (\mathcal{S}, \psi)} \qquad \frac{(\mathcal{S}, \varphi \leftrightarrow \psi)}{(\mathcal{S}, \varphi \rightarrow \psi), (\mathcal{S}, \psi \rightarrow \varphi)}$$

$$\frac{(\mathcal{S}, \neg(\varphi \vee \psi))}{(\mathcal{S}, \neg\varphi), (\mathcal{S}, \neg\psi)} \qquad \frac{(\mathcal{S}, \neg(\varphi \wedge \psi))}{(\mathcal{S}, \neg\varphi) \mid (\mathcal{S}, \neg\psi)}$$

$$\frac{(\mathcal{S}, \neg(\varphi \rightarrow \psi))}{(\mathcal{S}, \varphi), (\mathcal{S}, \neg\psi)} \qquad \frac{(\mathcal{S}, \neg(\varphi \leftrightarrow \psi))}{(\mathcal{S}, \neg(\varphi \rightarrow \psi)) \mid (\mathcal{S}, \neg(\psi \rightarrow \varphi))}$$

$$\frac{(\mathcal{S}, \neg\neg\varphi)}{(\mathcal{S}, \varphi)} \qquad \frac{(\mathcal{S}, \varphi), (\mathcal{S}', \neg\varphi)}{(\mathcal{S} \cup \mathcal{S}', \bot)}$$

There are three aspects to note:

- The right rule on the last line specifies the support for the closure of a branch of the semantic tableau,
- More than one support for the closure of a branch may be derived. Here we are interested in every support for a branch closure.

– For an element $(\mathcal{S}, \varphi)$ of a tableau node, unlike an argument defined by Definition 2, $\hat{\mathcal{S}} \vdash \varphi$ need not hold.

Four additional tableau rules are used for predicate logic.

$$\frac{(\mathcal{S}, \forall x\ \varphi)}{(\mathcal{S}, \varphi[^{x}/_{t}])} \qquad \frac{(\mathcal{S}, \exists x\ \varphi)}{(\mathcal{S}, \varphi[^{x}/_{c}])}$$

$$\frac{(\mathcal{S}, \neg(\forall x\ \varphi))}{(\mathcal{S}, \neg\varphi[^{x}/_{c}])} \qquad \frac{(\mathcal{S}, \neg(\exists x\ \varphi))}{(\mathcal{S}, \neg\varphi[^{x}/_{t}])}$$

Here t can be any term that occurs in the current node, and c must be a new constant not yet occurring the current node of the argumentation tableau. Since t can be any term, the corresponding rule can be applied more than once for the same proposition.

If an argumentation tableau closes, we can determine the support(s) for the closure.

Definition 5. *Let an argumentation tableau $\mathcal{T}$ with n leaf nodes: $\Lambda_1, \ldots, \Lambda_n$.*

– *The argumentation tableau is* closed *iff for every leaf Λ_i there is an argument $(\mathcal{S}_i, \bot) \in \Lambda_i$.*
– *A support for a tableau closure is defined as:* $\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}_i$ *where* $(\mathcal{S}_i, \bot) \in \Lambda_i$.

Note that a leaf of a closed tableau may contain more than one argument of the form $(\mathcal{S}', \bot)$. Therefore, there can be multiple supports for the closure of the tableau. In order to determine every possible $(\mathcal{S}', \bot)$, the leafs of the closed tableau must also be saturated. It may be impossible to determine saturated leafs in case of predicate logic.

Proposition 1. *Let $\mathcal{L}$ be the language of propositional or predicate logic, let the $\Sigma \subseteq \mathcal{L}$, and let $\mathcal{T}$ be an argumentation tableau. Then,*

1. *If $\mathcal{S}$ is a support for the closure of the tableau $\mathcal{T}$ with root node $\Gamma_0 = \{(\{\sigma\}, \sigma) \mid \sigma \in \Sigma\}$, then $\mathcal{S} \subseteq \Sigma$ is inconsistent.*
2. *If $\mathcal{S} \subseteq \Sigma$ is a minimal inconsistent set, then there exists a tableau $\mathcal{T}'$ which extends the tableau $\mathcal{T}$ and $\mathcal{S}$ is a support for the closure of $\mathcal{T}'$.*

Proof. A version of the paper with all proofs can be found at:
https://dke.maastrichtuniversity.nl/nico.roos/bnaic2020proofs.

A standard semantic tableau uses refutation to prove a conclusion. The support $\mathcal{S}$ for a closure of an argumentation tableau can be used for the same purpose. Since $\mathcal{S}$ is inconsistent for any $\sigma \in \mathcal{S}$, $\mathcal{S}\backslash\sigma \models \neg\sigma$. Hence, to prove φ and identify a corresponding argument, we add $(\{\neg\varphi\}, \neg\varphi)$ to the root Γ_0 of the tableau. If the tableau closes and if the support $\mathcal{S}$ of an inconsistency contains $\neg\varphi$, then we can construct an argument $(\mathcal{S}\backslash\neg\varphi, \varphi)$. To keep track of the proposition we try to refute, we put a question-mark behind the proposition in the support $(\{\neg\varphi?\}, \neg\varphi)$. The element $(\{\neg\varphi?\}, \neg\varphi)$ that we add to the root node, is called a *test*.

Corollary 1. *Let $\Sigma \subseteq \mathcal{L}$ be the initial information and let $\varphi \in \mathcal{L}$ be the proposition for which we search supporting arguments. Moreover, let $\mathcal{S}$ be the support for a tableau closure of a tableau $\mathcal{T}$ with root $\Gamma_0 = \{(\{\sigma\}, \sigma) \mid \sigma \in \Sigma\} \cup (\{\neg\varphi?\}, \neg\varphi)$.*

1. *If $\mathcal{S}$ is the support for a tableau closure of a tableau $\mathcal{T}$ and $\mathcal{S}$ contains a single test $\neg\varphi?$, then $\mathcal{S}\backslash\neg\varphi? \vdash \varphi$.*
2. *If $\mathcal{S}' \subseteq \Sigma$ is a minimal set such that $\mathcal{S}' \vdash \varphi$, then there exists a tableau $\mathcal{T}'$ which extends the tableau $\mathcal{T}$ and $\mathcal{S}' \cup \{\varphi?\}$ is a support its closure.*

It can be beneficial if we can derive multiple conclusions simultaneously. The argumentation tableau offers this possibility by simply adding several tests to the root node. After deriving a support $\mathcal{S}$ for a tableau closure, we check whether the support contains multiple tests. If it does, it can be ignored. We are interested in supports with zero or one test. The latter provides arguments for conclusion of interest, and the former enables us to handle with inconsistencies in the initial information. For instance Roos [16, 17] proposes to resolve the inconsistencies by formulating undercutting arguments for the least preferred propositions in $\mathcal{S}$ given a partial preference order $<$ (which can be empty).

Definition 6. *Let $\mathcal{S}$ be a support without tests for the tableau closure.*
For every $\sigma \in \min_< \mathcal{S}$, $(\mathcal{S}\backslash\sigma, \textbf{not } \sigma)$ is an undercutting argument of σ.

Other names that can be found in the literature for this form of undercutting attack are: *premise attack* and *undermining* [15]. The derivation of arguments for conclusions and undercutting arguments to resolve inconsistencies is related to [3, 4, 9, 17, 22].

4 Defeasible Rules

The argumentation tableau presented in the previous section enables us to derive deductive arguments. It does not support arguments containing defeasible rules. Here we will extend the argumentation tableau in order to derive arguments as defined in Definition 2.

The support of the argument defined in Definition 2 is a tree consisting of alternating deductive and defeasible transitions. The root of the tree is the conclusion / claim supported by the argument. For instance,

$$A : \begin{array}{l} p \vee q \\ \neg q \end{array} \Big| \vdash p \rightsquigarrow r \vdash r \rightsquigarrow s \vdash s$$

The support of the deductive transitions can be determined by the basic argumentation tableau described in the previous section by adding the antecedent of a defeasible rule as a test to the root of the argumentation tableau. Since we do not know which antecedents of defeasible rules will be supported, we add all of them as tests to the root Γ_0.

Next we extend the root of the tableau with the consequent of a defeasible rule after determining a support for its antecedent from a closed tableau.

Definition 7. *Let $\mathcal{T}$ be a tableau with root Γ_0. Moreover, let $\mathcal{S}$ be the support for the antecedent φ of the rule $\varphi \rightsquigarrow \psi \in \overline{D}$ determined by the tableau $\mathcal{T}$ where $(\{\neg\varphi?\}, \neg\varphi) \in \Gamma_0$.*

Then we can extend the root Γ_0 with the argument $(\{(\mathcal{S}, \varphi \rightsquigarrow \psi)\}, \psi)$.

To give an illustration, consider the initial information $\Sigma = \{p \vee q, \neg q\}$ and the defeasible rules $D = \{p \rightsquigarrow r, r \rightsquigarrow s\}$. We are interested in an argument for the conclusion s. We start constructing the tableau shown on the left in Figure 1. The support for the closure of this tableau is: $\{p \vee q, \neg q, \neg p?\}$ implying the argument $(\{p \vee q, \neg q\}, p)$. We may therefore add the consequence r of the defeasible rule $p \rightsquigarrow r$ with the support $\{(\{p \vee q, \neg q\}, p \rightsquigarrow r)\}$ to the root of the tableau and continue rewriting the tableau. This results in the tableau shown on the right in Figure 1.

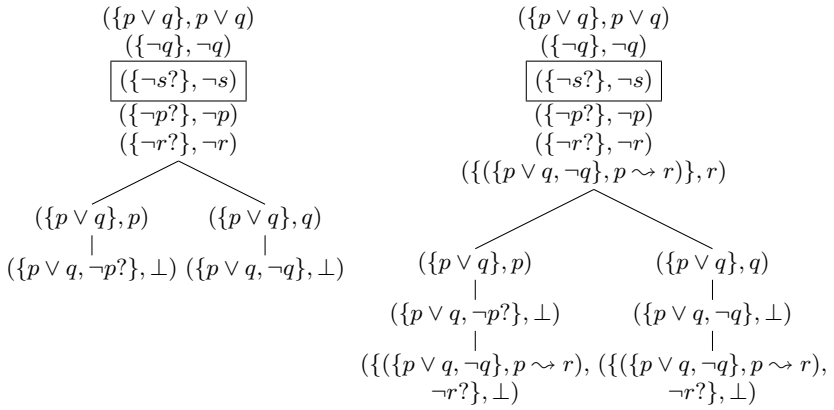


Fig. 1. Deriving defeasible arguments 1

The support for the new closure of the tableau shown on the right in Figure 1 is: $\{(\{p \vee q, \neg q\}, p \rightsquigarrow r), \neg r?\}$ implying the argument $(\{(\{p \vee q, \neg q\}, p \rightsquigarrow r)\}, r)$. We may therefore add the consequence s of the defeasible rule $r \rightsquigarrow s$ with the support $\{(\{(\{p \vee q, \neg q\}, p \rightsquigarrow r)\}, r \rightsquigarrow s)\}$ to the root of the tableau and continue rewriting the resulting tableau as shown in Figure 2. The support for the closure of the tableau as shown in Figure 2 is:

$$\{(\{(\{p \vee q, \neg q\}, p \rightsquigarrow r)\}, r \rightsquigarrow s), \neg s?\}$$

implying the desired argument

$$(\{(\{(\{p \vee q, \neg q\}, p \rightsquigarrow r)\}, r \rightsquigarrow s)\}, s)$$

4.1 Predicate Logic

The construction of an argumentation tableau for predicate logic extended with defeasible rules is the same as the above described argumentation tableau for

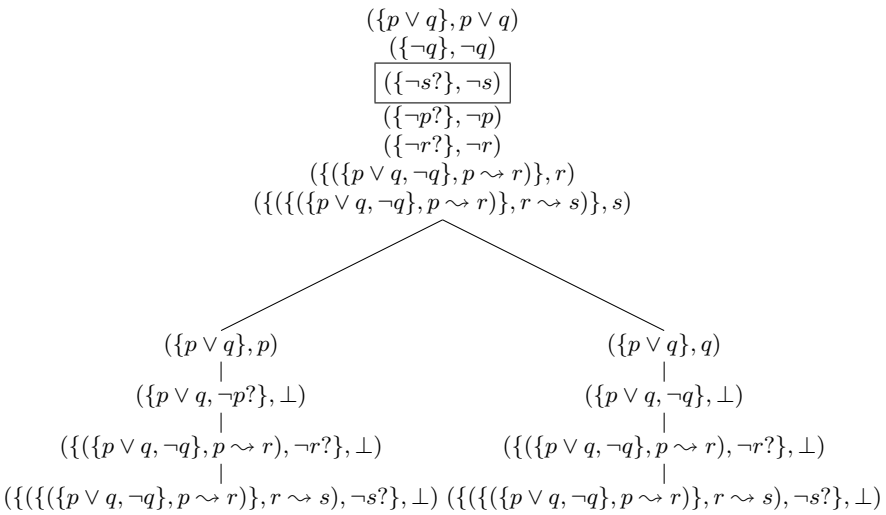


Fig. 2. Deriving defeasible arguments 2

propositional logic with defeasible rules. We should in principle add every ground instance of the negated antecedent of each rule $\varphi(\mathbf{t}) \rightsquigarrow \psi(\mathbf{t}) \in \overline{D}$ as a test to the root of the tableau. That is, we should add the set of tests

$$\{(\{\neg\varphi(\mathbf{t})?\}, \neg\varphi(\mathbf{t})) \mid \varphi(\mathbf{t}) \rightsquigarrow \psi(\mathbf{t}) \in \overline{D}, \mathbf{t} \in \mathcal{G}\}$$

to the root of the tableau. If functions are used, this set of tests will be infinite, and therefore adding all ground instances is not practically feasible. Instead, we may limit ourselves to the ground instances that are present in the current tableau. So, while expanding the tableau, more ground instance may be added.

4.2 Correctness and Completeness

We can proof that the argumentation tableau determines exactly the same set of arguments as those defined in Definition 2. First, we prove a proposition similar to Proposition 1

Proposition 2. *Let $\mathcal{L}$ be the language of propositional or predicate logic, let the $\Sigma \subseteq \mathcal{L}$, let D be a set of defeasible rules over $\mathcal{L}$, and let $\Gamma_0 = \{(S_i, \psi_i)\}_{i=1}^n$ be the root node of the tableau $\mathcal{T}$. Then,*

1. *If $\mathcal{S}$ is a support for the closure of the tableau $\mathcal{T}$, then $\hat{\mathcal{S}} \subseteq \Sigma$ is inconsistent.*
2. *If $\hat{\mathcal{S}} \subseteq \Sigma$ is a minimal inconsistent set, then $\mathcal{S}$ is a support for the closure of the tableau $\mathcal{T}$.*

Theorem 1. *If A is a minimal argument according to Definition 2, then A can be derived by an argumentation tableau. If the argument A can be derived by an argumentation tableau, then A is an argument according to Definition 2.*

5 Reasoning by Cases

Reasoning by cases addresses the derivation of conclusions in the context of uncertainty. Uncertainty described by disjunctions results in multiple cases. Each case is a possible description of the world. If the same conclusion is derived in each case, then that conclusion will certainly hold in the case describing the world. The use of defeasible rules to derive new conclusions in a case should make no difference despite that the arguments supporting the conclusions might defeat other arguments.

5.1 Cases in an argumentation tableau

If we ignore the *tests* that we add to the root of an argumentation tableau, then the construction of a tableau can be viewed as the construction of all cases implied by the available information. Ignoring the tests, each open branch describes one case implied by the available disjunctive information. If a case describes the world, additional information may eliminate all other cases and a defeasible rule should be applied as described in the previous section.

The use of defeasible rules in a case implies that we should extend a leaf of the argumentation tableau with the consequence of a defeasible rule whenever the leaf entails the antecedent this rule. We cannot test whether a leaf entails the antecedent of a defeasible rule by adding the antecedent as a test to the root of the tableau. We should add the antecedent to the leaf. Preferably the leaf is saturated because a possibly successful test may fail if we add it too early. To give an illustration, consider $\Sigma = \{p \vee q\}$ and $D = \{p \rightsquigarrow r, q \rightsquigarrow r\}$. If we add the tests $(\{\neg p?\}, \neg p)$ and $(\{\neg q?\}, \neg q)$ to the root of the tableau, both tests will fail because there is no support for a tableau closure with only one test. If however we first rewrite $p \vee q$ and then add the test to the resulting leafs, in each branch we will derive a support for a closure that enables us to add the consequence of the corresponding rule. The two cases are illustrated by the two tableaux in Figure 3.

The example illustrates that adding the test is a strategic choice, which can be dealt with through search. We add a test for the negated antecedent of a rule to a current leaf and try to close all resulting branches starting from the leaf. If we cannot close all these branches, we backtrack to the leaf and remove the test. Using such a search process is of course not a very efficient solution.

Instead of adding tests for the antecedents of defeasible rules, we can check whether the current leaf of a branch of a tableau entails the antecedent. This works fine for propositional logic but raises a problem for predicate logic. If the antecedent of a rule contains a universal claim; i.e., a universally quantified proposition that must be true or an existentially quantified proposition that must be false, then entailment is not decidable because we do not know all the objects over which we have to quantify. So we should restrict the defeasible rules to those that do not contain universal claims in the antecedent. This restriction implies that we cannot state that *a Student that Passes all Exams normally*

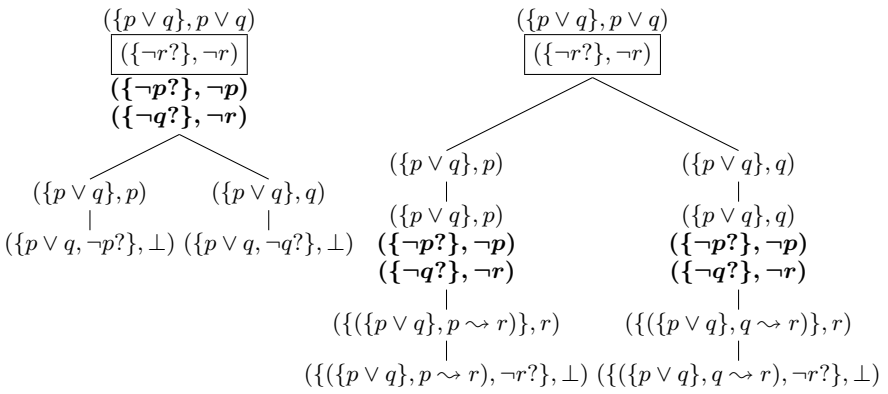


Fig. 3. Reasoning by cases.

receives a Diploma: $S(x) \wedge \forall[E(y) \rightarrow P(x, y)] \rightsquigarrow D(x)$. This even holds if the exams have been specified explicitly: $\forall y[E(y) \leftrightarrow y = e_1 \vee \cdots \vee y = e_n]$.

A possible solution for this restriction is a first order logic that uses binary quantifiers in combination with a special specification of the ground terms for which a predicate is true: $E = \{e_1 \ldots e_n\}$ and $S(x) \wedge \forall E(y)[P(x, y)] \rightsquigarrow D(x)$. However, if we wish to stay in the domain of standard predicate logic, we should rely on the above described search process.

5.2 How to reason by cases with defeasible information

There have been a few proposals how to introduce reasoning by cases in argumentation systems [2, 6, 13, 19]. Unfortunately, there is no consensus on the correct conclusion(s) when reasoning by cases using defeasible information. Here we propose that *the (defeasible) conclusions supported in a case by defeasible information must be the same as when uncertainty is eliminated by additional information*. This principle implies that we only need to eliminate cases (through additional information) in which the antecedent of a defeasible rule does not hold. A case can therefore have sub-cases. To give an illustration, consider the information $\Sigma = \{\neg(p \wedge q), r \vee s, t\}$ and the defeasible rules $D = \{r \rightsquigarrow p, t \rightsquigarrow q\}$. The defeasible rule $r \rightsquigarrow p$ is applicable in the case $\{\neg(p \wedge q), r, t\}$. This case has two sub-cases, $\{\neg p, r, t\}$ and $\{\neg q, r, t\}$. An inconsistency can be derived in the case $\{\neg(p \wedge q), r, t\}$ and the set of last rules involved in the inconsistency is: $\{r \rightsquigarrow p, t \rightsquigarrow q\}$.

Before addressing the technical details of reasoning be cases in using an argumentation tableau, we will first briefly review proposals made in the literature.

– Pollock's argumentation system OSCAR [12, 13] is an example of an argumentation system that allows for suppositional reasoning, and is therefore capable of reasoning by cases. Pollock does not explicitly discuss which conclusions should be supported when using reasoning by cases with defeasible

rules. His definition of rebutting attack [12] implies that a suppositional argument can only be defeated by (1) suppositional arguments of the same case, and (2) by arguments that do not depend on the considered cases. A suppositional argument cannot defeat an argument that does not depend on any case. As argued in [19], this restriction may result in incorrect conclusions.
- Bodanza [6] adapts OSCAR by allowing that a suppositional argument defeats an argument that does not depend on any case. However, Bodanza changes the interpretation of the $\neg$-operator. $\neg\alpha$ is interpreted as: "α is not an alternative" when reasoning by cases.
- Recently, the framework for structured argumentation ASPIC$^+$ [11, 15] has been extended in order to enable reasoning by cases [2]. The authors introduce hypothetical sub-arguments to handle the cases. An argument can attack a hypothetical sub-argument but not vice versa. Hypothetical sub-arguments can only attack other hypothetical sub-arguments.

The first and the last approach above result in counter-intuitive conclusions in the following example.

> Harry and Draco are involved in a fight and therefore are punishable. However, if someone involved in a fight, acted in self-defense, then he or she is not punishable. Witnesses state that either Harry or Draco acted in self-defense.

The first and last approach above support the conclusion that both Harry and Draco are punishable, while we would expect that only one of them is punishable. Our proposal that *conclusions supported in a case by defeasible information must be the same as when uncertainty is eliminated by additional information* avoids the counter-intuitive conclusion. However, it introduces a technical issue, which will be discussed in the next subsection.

5.3 Local tableau closures

Reconsider the above example with information $\Sigma = \{\neg(p \wedge q), r \vee s, t\}$ and defeasible rules $D = \{r \rightsquigarrow p, t \rightsquigarrow q\}$. We can use the information and the rules to construct the tableau on the left in Figure 4. If we eliminate the right most branch by adding the information $\neg s$, we get a tableau as described in Section 4, and the set of last rules for the derived inconsistency is: $\{r \rightsquigarrow p, t \rightsquigarrow q\}$. It is not difficult to determine the same inconsistency in left tableau in Figure 4.

It is also possible to construct the tableau on the right in Figure 4 using the same information. Here it is more difficult to determine the set of last rules involved in the inconsistent case.

The key to identify an inconsistent case is by checking whether all alternatives implied by the propositions $\hat{\mathcal{S}}$ of a closed branch with support $\mathcal{S}$ for the closure, are also closed. Consider the closed left branch in the left and the right tableau in Figure 4. The support $\mathcal{S} = \{\neg(p \wedge q), (\{r \vee s\}, r \rightsquigarrow p)\}$ for the closure is based on one of the two cases implied by $\neg(p \wedge q)$, namely the case in which $\neg p$

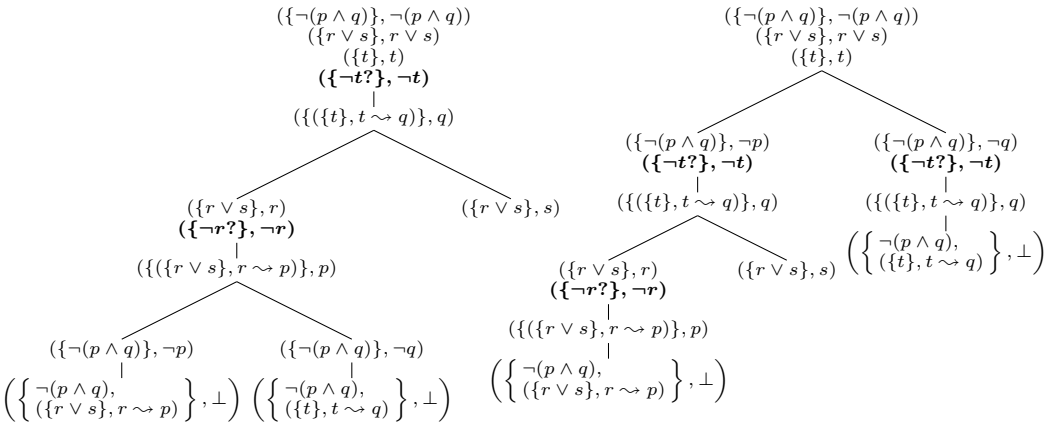


Fig. 4. Local tableau closure.

holds. It is possible that the other case in which $\neg q$ holds, is consistent. The case $\{\neg(p \wedge q), r\}$ can only be inconsistent if the case $\{\neg q, r\}$ is inconsistent too.

To determine whether a case is inconsistent, which we will call a *local tableau closure*, we need to consider all cases implied by a set of propositions $\hat{\mathcal{S}}$ where $\mathcal{S}$ is the support of a branch closure. Since these cases can be spread over the whole tableau, we will propagate the support for branch closures towards the root of the tableau. This enables us to check for a proposition involved in a leaf closure whether all cases implied by the proposition are closed.

Definition 8. *Let $\mathcal{T}$ be an argumentation tableau with root Γ_0 and with leaf nodes: $\Lambda_1, \ldots, \Lambda_n$. Moreover, let $\Lambda_{i_1}, \ldots, \Lambda_{i_k}$ be the closed leaf nodes. We propagate the support for the closure of a leaf toward the root of the tableau.*

- *If the argument $(\mathcal{S}, \eta)$ was rewritten in a node Γ and resulted in one child node Γ', then add every $(\mathcal{S}, \bot) \in \Gamma'$ to Γ.*
- *If the argument $(\mathcal{S}, \eta)$ was rewritten in a node Γ and resulted in more than one child node $\Gamma_1, \ldots, \Gamma_m$, then add $(\bigcup_{i=1}^{m} \mathcal{S}_i, \bot)$ with $(\mathcal{S}_i, \bot) \in \Gamma_i$ and $\mathcal{S} \subseteq \mathcal{S}_i$, to Γ.*
- *If the argument $(\mathcal{S}, \eta)$ was rewritten in a node Γ and resulted in more than one child node $\Gamma_1, \ldots, \Gamma_m$, then add every $(\mathcal{S}_i, \bot) \in \Gamma_i$ such that $\mathcal{S} \not\subseteq \mathcal{S}_i$, to Γ.*

Every $(\mathcal{S}, \bot) \in \Gamma_0$ represents a local tableau closure.

When we apply the procedure in this definition to the above example, we get the tableau shown in Figure 5. The tableau supports the local closure that we expect.

We can prove that Definition 8 guarantees that supports for local closures represent inconsistent cases.

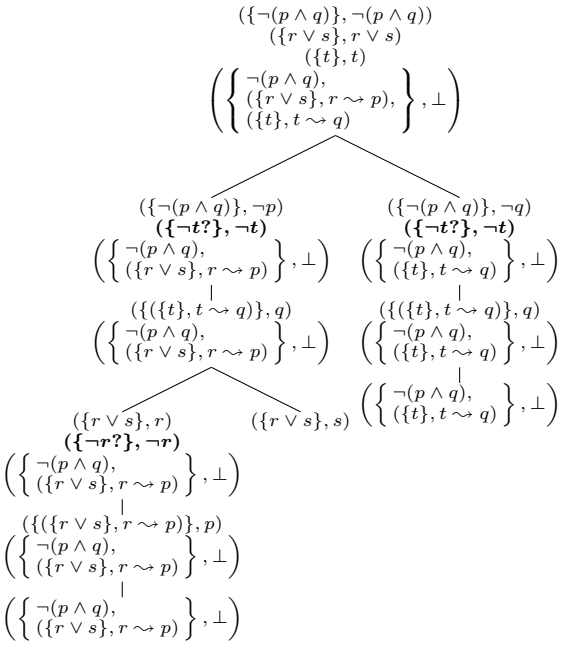


Fig. 5. The support for a local tableau closure.

Proposition 3. *If $\mathcal{S}$ is the support for the local closures of a tableau, then $\hat{\mathcal{S}} \vdash \bot$.*

We can also prove that inconsistent cases can be identified through supports for local closures.

Proposition 4.
Let $\{(\{\eta_1\}, \eta_1), \ldots, (\{\eta_m\}, \eta_m), (\mathcal{S}_1, \mu_1), \ldots, (\mathcal{S}_n, \mu_n)\}$ be a case that is considered by the argumentation tableau. If $\{\eta_{i_1}, \ldots, \eta_{i_k}, \mu_{i_1}, \ldots, \mu_{i_l}\}$ is a minimal inconsistent set, then $\mathcal{S} = \{(\{\eta_{i_1}\}, \eta_{i_1}), \ldots, (\{\eta_{i_k}\}, \eta_{i_k}), (\mathcal{S}_{i_1}, \mu_{i_1}), \ldots, (\mathcal{S}_{i_k}, \mu_{i_k})\}$ is a support for a local closure.

5.4 Mutually exclusive cases

There is one last issue concerning reasoning by cases. The tableau rule $\frac{(\mathcal{S}, \varphi \vee \psi)}{(\mathcal{S}, \varphi) | (\mathcal{S}, \psi)}$ does not guarantee that cases are mutually exclusive. The applying this tableau rule results in two children representing two cases. Both cases may support a conclusion η. This conclusion is not justified if η does not hold when both φ and ψ are true. As an illustration, suppose that a party will be great if Harry or Ron will attend it, but not if both will attend (because Harry and Ron have a quarrel). Here the case that Harry attends the party and whether Ron attends is unknown, is not the same as drawing a conclusion in the absence of more specific

information. The disjunction implies that Ron might attend the party too. The solution to this issue is to ensure that the tableau only contains cases that are mutually exclusive. We address this problem by adapting three tableau rules.

$$\frac{(\mathcal{S}, \varphi \vee \psi)}{(\mathcal{S}, \varphi \wedge \neg\psi) \mid (\mathcal{S}, \varphi \wedge \psi) \mid (\mathcal{S}, \neg\varphi \wedge \psi)} \qquad \frac{(\mathcal{S}, \varphi \rightarrow \psi)}{(\mathcal{S}, \neg\varphi \wedge \neg\psi) \mid (\mathcal{S}, \neg\varphi \wedge \psi) \mid (\mathcal{S}, \varphi \wedge \psi)}$$

$$\frac{(\mathcal{S}, \neg(\varphi \wedge \psi))}{(\mathcal{S}, \neg\varphi \wedge \psi) \mid (\mathcal{S}, \neg\varphi \wedge \neg\psi) \mid (\mathcal{S}, \varphi \wedge \neg\psi)}$$

Using these adapted tableau rules we will consider three mutually exclusive cases given the information that Harry or Ron will attend the party. In two cases the party will be great and in one case it will not.

6 Conclusion

This paper investigated the possibility of using the semantic tableau method to derive arguments for claims / conclusions. We conclude that it is possible to define an argumentation tableau that provides the arguments supporting conclusions in case of propositional and predicate logic. If the initial information is inconsistent, undercutting arguments can also be derived for resolving the inconsistencies. We further conclude that an argumentation tableau can provide arguments supporting conclusions if propositional and predicate logic are extended with defeasible rules. Arguments for inconsistencies, covering rebutting attacks, can be resolved by deriving undercutting arguments for defeasible rules. Our last conclusion is that an argumentation tableau enables reasoning by cases and that conclusions supported by reasoning by cases are intuitively plausible.

Further research can be done on (*i*) efficiently implementing an argumentation tableau, and (*ii*) adapting the argumentation tableau to other logics.

References

1. Baroni, P., Giacomin, M., Guida, G.: Scc-recursiveness: a general schema for argumentation semantics. Artificial Intelligence **168**, 162–210 (2005)
2. Beirlaen, M., Heyninck, J., Straßer, C.: Reasoning by cases in structured argumentation. In: Proceedings of the Symposium on Applied Computing. pp. 989–994. SAC '17 (2017)
3. Besnard, P., Hunter, A.: Practical first-order argumentation. In: Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI). pp. 590–595 (2005)
4. Besnard, P., Hunter, A.: Argumentation based on classical logic. In: Simari, G., Rahwan, I. (eds.) Argumentation in Artificial Intelligence, pp. 133–152. Springer (2009)
5. Beth, E.W.: Formal Methods An Introduction to Symbolic Logic and to the Study of Effective Operations in Arithmetic and Logic. Synthese library. D. Reidel Publ. Comp.

6. Bodanza, G.: Disjunctions and specificity in suppositional defeasible argumentation. Logic Journal of the IGPL **10**(1), 23–49 (2002)
7. Caminada, M.: Semi-stable semantics. In: Proceedings of the 1st Conference on Computational Models of Argument (COMMA 2006). vol. 144 of Frontiers in Artificial Intelligence and Applications. IOS Press (2006)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**, 321–357 (1995)
9. Dung, P.M., Kowalski, R.A., Toni, F.: Assumption-based argumentation. In: Rahwan, I., Simari, G. (eds.) Argumentation in Artificial Intelligence, pp. 199–218. Springer (2009)
10. Dung, P., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artificial Intelligence **171**, 642–674 (2007)
11. Modgil, S., Prakken, H.: The ASPIC$^+$ framework for structured argumentation: a tutorial. Argument & Computation **5**, 31–62 (2014)
12. Pollock, J.L.: A theory of defeasible reasoning. International Journal of Intelligent Systems **6** (1991)
13. Pollock, J.L.: How to reason defeasibly. Artificial Intelligence **57**, 1–42 (1992)
14. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Gabbay, D.M., Guenthner, F. (eds.) The Handbook of Philosophical Logic, pp. 219–318. Springer Netherlands (2002)
15. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument & Computation **1**(2), 93–124 (2010)
16. Roos, N.: Preference logic: a logic for reasoning with inconsistent knowledge. Tech. Rep. 89-53, Delft University of Technology, Faculty of Technical Mathematics and Informatics. ISSN 0922-5641 (1989)
17. Roos, N.: A logic for reasoning with inconsistent knowledge. Artificial Intelligence **57**, 69–103 (1992)
18. Roos, N.: On resolving conflicts between arguments. Tech. rep., TR-CTIT-97-37 Centre for Telematics and Information Technology, University of Twente, Enschede (1997)
19. Roos, N.: On resolving conflicts between arguments. Computational Intelligence **16**, 469–497 (2000)
20. Roos, N.: Preferential model and argumentation semantics. In: Proceedings of the 13th International Workshop on Non-Monotonic Reasoning (NMR-2010) (2010)
21. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artificial Intelligence **53**, 125–157 (1992)
22. Toni, F.: A tutorial on assumption-based argumentation. Argument & Computation **5**(1), 89–117 (2014)
23. Toulmin, S.: The uses of argument. Cambridge University Press (1958)
24. Verheij, B.: Two approaches to dialectical argumentation: Admissible sets and argumentation stages. In: In Proceedings of the biannual International Conference on Formal and Applied Practical Reasoning (FAPR) workshop. pp. 357–368 (1996)
25. Vreeswijk, G.: Abstract argumentation systems. Artificial Intelligence **90**, 225–279 (1997)

'Thy algorithm shalt not bear false witness': An Evaluation of Multiclass Debiasing Methods on Word Embeddings

Thalea Schlender and Gerasimos Spanakis[0000−0002−0799−0241]

Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, Netherlands

Abstract. With the vast development and employment of artificial intelligence applications, research into the fairness of these algorithms has been increased. Specifically, in the natural language processing domain, it has been shown that social biases persist in word embeddings and are thus in danger of amplifying these biases when used. As an example of social bias, religious biases are shown to persist in word embeddings and the need for its removal is highlighted. This paper investigates the state-of-the-art multiclass debiasing techniques: Hard debiasing, SoftWEAT debiasing and Conceptor debiasing. It evaluates their performance when removing religious bias on a common basis by quantifying bias removal via the Word Embedding Association Test (WEAT), Mean Average Cosine Similarity (MAC) and the Relative Negative Sentiment Bias (RNSB). By investigating the religious bias removal on three widely used word embeddings, namely: Word2Vec, GloVe, and ConceptNet, it is shown that the preferred method is ConceptorDebiasing. Specifically, this technique manages to decrease the measured religious bias on average by 82,42%, 96,78% and 54,76% for the three word embedding sets respectively.

Keywords: Natural Language processing · Word Embeddings · Social Bias

1 Introduction

In recent years, there have been rapid advances in artificial intelligence and the accompanying vast development of machine learning applications. With the increased wide spread (commercial) employment of such applications it has become increasingly more vital to ensure their transparency, fairness and equality. Recent investigations of various application domains have shown that many of these applications exhibit several social biases endangering their fairness [16]. Social biases describe the discrimination of certain identity groups based on, for example, their gender, race or religion. When social biases persist in machine learning applications, they run the danger of amplifying these biases. For instance, regarding social bias against minority groupds, it was found that these were recognized considerably less [6]. To illustrate the real world consequences

which minority group members face through biased algorithms, consider the use of these face /voice applications in sensitive areas such as medical diagnosis or the justice system. In cases like these, "the use of biased information could entail an extended and undeserved period of incarceration, which unjustly affects those who are arrested and possibly ruins the lives of their families" (p.7, [6]). With respect to a medical application, "consider a revolutionary test for skin cancer that does not work on African Americans" (p.1, [14]).

Biases inherent in our society are, thus, perpetuated in the machine learning models, recorded by the model's outcomes and, hence, threaten to treat various groups differently. To rectify the unequal treatment, the origin of biases in artificial intelligence needs to be examined and, consequently, removed. These biases in data driven applications may have myriad causes. One cause is the gathering of the data that is primarily done or planned by humans, which causes the data to be subject to similar biases as humans have. Moreover, the gathering process favours easy accessible and quantifiable data [15], which may favour certain societal groups over others. Further, biases are captured in the under- / over-representation of societal groups in the dataset, which makes the complete data not representative of the end users anymore [15]. Another origin of bias is data directly containing sensitive attributes, such as race or religion, or any proxy features for these. These proxy features may be well hidden, for instance a societal group may be represented in the post codes of communities. With the encoding of sensitive information, an algorithm can learn wrong causal inferences concerning these which can be hard to identify [15].

The origins of bias mentioned above can be present in many representations of data. To provide an elaborate analysis, this paper will henceforth tend to textual data solely. To process textual data for an application, the data must be represented numerically. This is done via word embeddings, which attempt to capture the meaning and semantic relationships of a word and translate these to a real valued vector. Since word embeddings are learnt from possibly biased data, word embeddings themselves may contain biases, which could ripple through an application. Having outlined why the mitigation of these biases is vital and having introduced the domain of biased word embeddings, this paper will review work on analysis and mitigation of biased word embeddings, before presenting and evaluating various state-of-the-art post processing approaches to the mitigation of the found biases. Specifically, the attempted removal of multiclass social biases in three word embeddings is quantified on geometrical as well as on downstream evaluation metrics.

In order to highlight the results, the problem of religious bias is taken as a novel example for multi-class social bias. By doing so this paper aims to answer following research questions:

- To what extent are Religious biases, as an example for social bias, present in widely used word embeddings?
- How do state-of-the-art multiclass debiasing techniques compare geometrically?

– How do state-of-the-art multiclass debiasing techniques compare considering the discrimination of a downstream application?

To address which state-of-the-art debiasing technique performs religious debiasing the best, an extensive background on social biases in word embeddings is given. The evaluation metrics this paper uses to access performance are explained, before the debiasing techniques examined are illustrated. This paper, then, highlights the need for religious debiasing by showing its presence in a word embedding. Consequently a common base for the analysis of bias removal is established to compare the debiasing methods. Finally, this paper discusses the performance of the debiasing techniques and based on this evaluation, advises the use of one.

2 Background

Social biases have been found in popular, widely used word embeddings such as GloVe [18] or word2Vec [13], [3]. Specifically, gender biases have been found to persist by creating simple analogies, which have led to the example "Man is to Computer Programmer as Woman is to Homemaker" [3], [1]. This analogy clearly shows that the word embeddings have captured gender bias with regards to occupation, which may cause disruption in, e.g. a CV-Scanning application. Similarly, the multi-class racial bias in word embeddings has led to other biased analogies [11] being coined. Sweeney and Najafan have also shown that multi-class bias based on nationality or religion is present in word embeddings, which endangers specific identity groups to be treated differently [21].

Social biases have, therefore, been proven to likely exist within word embeddings. As mentioned before (1), biases in data driven artificial intelligence and ,thus, word embeddings have many causes, especially related to the bias present in the data used. Papakyriakopoulos, Hegelich, Serrano, and Marco find that biases in word embeddings are closely related to the input training data [17]. In fact, even when the text used for training was written for a "formal and controlled environment like Wikipedia, [it] result[ed] in biased word embeddings"(p.455, [17]).

A strong cause for bias in textual data is the more frequent co-occurrence of particular words to the identity terminology of one group rather than the other(s). Word embedding algorithms typically take co-occurrences as an indicator of context and semantic relationships. Thus, the word embeddings learn a stronger association between, for example, 'woman' and 'nurse' than 'man' and 'nurse'. This association, however, is an example of a stereotype, which should ideally not be captured in the artificial intelligence applications. Garg, Schiebinger, Jurafsky and Zou confirm that word embeddings "accurately capture both gender and ethnic occupation percentages" (p.3636, [4]).

The biases within word embeddings can amplify through an application, causing unfair results, which may influence actions in the real world. This, in turn, may lead to unequal treatment based on certain sensitive attributes and actively cause discrimination. Hence, it is vital to establish mitigation methods.

Debiasing methods may tend to different categories of biases. For instance, debiasing binary biases mitigates the unequal treatment of two groups based on a sensitive feature, and joint debiasing mitigates biases based on various sensitive attributes simultaneously. This paper demonstrates a multi-class debiasing, which deals with bias across more than two groups, by considering three religious groups, namely: Christianity, Islam, Judaism. The development of debiasing techniques is novel research, yet a few state-of-the-art approaches have been proposed. Following the notion that word embedding biases are a direct result of bias in the data, Brunet, Alkalay-Houlihan, Anderson, and Zemel have proposed a technique to track which segment of data is responsible for some bias [2]. It follows naturally that this can be applied as a debiasing technique by omitting these segments when training the word embedding model. Most debiasing techniques, however, concentrate on post-processing pre-trained word embeddings.

Bolukbasi, Chang, Zou, and Saligrama propose soft and hard debiasing as binary debiasing methods [1], which Manzini, Lim, Tsvetkov, and Black transfer into the multi-class domain [11]. Popovic, Lemmerich and Strohmaier expand these debiasing techniques further into SoftWEAT and hardWEAT, which also are applicable for joint debiasing [19]. Another joint multiclass debiasing approach is the Conceptor debiasing method by Karve, Ungar and Sedoc [9].

With the increased research into debiasing methods, Gonen and Goldberg [5] provide a critical view on the effectiveness of debiasing. The removal of bias in the techniques, such as hard debiasing, relies on the definition of the bias as being the projection onto a biased subspace. Gonen and Goldberg, however, believe that this is a mere indication of the presence of bias. Thus, although the debiasing methods may eliminate the bias projections, the bias is still captured within the geometry of supposedly neutralized words [5]. Hence, it is important to consider the quantification of bias removal critically.

In this paper, the multi-class debiasing methods, all mentioned above, namely Hard debiasing, SoftWEAT debiasing and Conceptor debiasing will be evaluated on different metrics in an attempt to quantify bias removal from geometrical and down stream perspectives. Previous work comparing debiasing techniques have evaluated their performance on merely one geometric metric quantifying bias [1], [11], [9], whereas this paper uses two geometric metrics, in addition to utilizing a downstream bias metric.

These metrics and debiasing techniques will now be introduced, before an investigation of religious bias, as an example of multiclass social bias, is conducted on a word embedding. Having established the need for religious debiasing, the bias removal will be conducted and analysed.

3 Methodology

3.1 Terminology

To aid in the explanation of the debiasing techniques and evaluation metrics, some definitions and terminologies are introduced first.

- A class C consists of a set of protected groups defined by some criteria, like religion or race.
- A subclass S_c then refers to a particular protected group within that class, such as Judaism when considering the religion class.
- An equality set E for a class is a set containing a term for each subclass, where all terms can be considered to denote an equivalent concept within each subclass. Thus, for instance, an equality set for C = religion with S_c = (Christianity, Islam, Judaism) could be (Church, Mosque, Synagogue).
- A target set T is a set of identity terms referring to a particular sub-class, thus inherently carrying bias. For Christianity this could include: {Church, Churches, Bible, Bibles, Jesus}
- An attribute set A contains sets of words referring to several topics, none of which should, in principle, be linked to the target set of a subclass, but that a target set of words may be associated to [19]. The aim of the debiasing methods is to remove this link. Examples for attribute sets are collections of words considered to be pleasant, or unpleasant, respectively or collections of words describing notions such as families, arts or occupations.

3.2 Bias Measurements Techniques

To quantify the bias removal, the three metrics introduced below are used. The first two metrics introduced evaluate the removal geometrically by considering the cosine distance of target and attribute sets, whereas the third highlights bias presence via a simple sentiment analysis application.

Word Embedding Association Test (WEAT) The standard evaluation of bias is the Word Embedding Association Test ($WEAT$) as established by Caliskan, Bryson, and Narayanan. It is widely used, for instance in [1] and [19], and it has been expanded, for instance, to the Sentence Encoder Association Test (SEAT) [12].

WEAT tests the association between one target and attribute set, relative to the association of the other target and attribute set in order to examine the null hypothesis that both target sets are equally similar to both attribute sets and not exhibiting any bias [3].

To perform WEAT, the mean cosine similarity of the target set T_1 to attribute sets A_1 and A_2 is compared to the mean cosine similarity of the target set T_2 to A_1 and A_2. The exact calculations for the test statistic $S(T_1, T_2, A_1, A_2)$ and the effect size d of the two attribute - target set pairs is given below. Let $s(w, A_1, A_2)$ be defined as in equation 1, where w is a given word vector:

$$s(w, A_1, A_2) = mean_{a_1 \in A_1} cos(\vec{w}, \vec{a}_1) - mean_{a_2 \in A_2} cos(\vec{w}, \vec{a}_2) \tag{1}$$

$$S(T_1, T_2, A_1, A_2) = \sum_{t_1 \in T_1} s(t_1, A_1, A_2) - \sum_{t_2 \in T_2} s(t_2, A_1, A_2), \tag{2}$$

The effect size d quantifies how distant these two associations of target and attribute pairs are. The closer the effect size d is to zero, the less distant the two associations are and thus, the less bias can be found between the target and attribute sets [3].

$$d = \frac{mean_{t_1 \in T_1} s(t_1, A_1, A_2) - mean_{t_2 \in T_2} s(t_2, A_1, A_2)}{std\text{-}dev_{w \in T_1 \cup T_2} s(w, A_1, A_2)} \tag{3}$$

It should be noted that bias here is defined on the relative distances.

Mean Average Cosine Similarity (MAC) WEAT as proposed by Caliskan et al. [3] provides a geometric interpretation of the distance between two sets of target words and two sets of attribute words.

The mean average cosine similarity (MAC) uses the intuition behind WEAT and applies this notion to a multiclass domain as proposed by Manzini et al. [11]. Instead of comparing the associations of one target set T_1 and an attribute set A_1, to the association of T_2 and A_2, MAC considers the association of one target set T_1 to all attribute sets A at one time.

The MAC metric is computed by calculating the mean over the cosine distances between an element t in a target set T to each element in an attribute set A, as seen in equation 4, in which the cosine distance is defined as $cos_{distance}(t, a) = 1 - cos(t, a)$. This is repeated for all elements in T to all attribute sets. The MAC then describes the average cosine distance between each target set and all attribute sets.

$$s_{MAC}(t, A_j) = \frac{1}{|A_j|} \sum_{a \in A_j} cos_{distance}(t, a) \tag{4}$$

Relative Negative Sentiment Bias (RNSB) The relative negative sentiment bias ($RNSB$) is an approach proposed by Sweeney and Najafan [21] in order to offer insights on the effect of biased word embeddings through downstream applications. Its framework involves training a logistic classifier to predict the positive or negative sentiment of a given word. The classifier is trained on supposedly unbiased sentiment words, which are encoded via the word embedding to be investigated. Sweeney and Najafan then encode identity terms and predict their respective negative sentiment probability. These results are used to form a probability distribution P. Intuitively, unbiased word embeddings would result in this probability distribution to be uniform, i.e. each class has equal probability of being classified as of negative sentiment. The RNSB is then defined as Kullback-Leibler divergence of P from the uniform distribution U [21].

3.3 Debiasing Techniques

These three metrics will be used to quantify the bias removal in the three debiasing techniques considered in this paper. Namely, these are Hard debaising, SoftWEAT and Conceptor debiasing.

Hard Debiasing Bolukbasi et al. [1] established two binary debiasing methods, namely: Soft and Hard debiasing, which Manizini et al. [11] then applied to the multiclass domain. These approaches mainly rely on two steps: The identification of a bias subspace, and the subsequent removal of that bias. The main difference between these two methods is the severity of bias removal.

The bias subspace identification utilizes equality sets E_i. For each set, the center of the set is computed and the distance of each term in the equality set to the center is considered. The subspace capturing the class is then found by examining the variance of each term. Bias removal is carried out by a 'neutralize and equalize' approach. The projection of words that are declared neutral onto the bias subspace is subtracted from their word vector. The identity words, however, rely on their bias component. Thus, in the equalization step, the terms within an equality set, are centralized and are each given an equal bias component.

SoftWEAT Debiasing Popovic et al. propose debiasing techniques SoftWEAT and hardWEAT [19], which borrow intuition from WEAT [3]. SoftWEAT expands the target set of each subclass by considering the n closest neighbours to all identity terms. Merely this set is then manipulated. To find the linear transformation to be applied, the attribute sets the target set of a subclass is biased against is found via WEAT and their respective null space vectors are calculated. The translation of the subclass embeddings is then taken from the null space vector, which decreases the WEAT score the most. The final transformation can be scaled by hyper-parameter λ.

Conceptor Debiasing Karve et al. developed the Conceptor debiasing post processing method [9]. The notion of this method is to generate a conceptor, as defined by Jaeger [8], to represent bias directions and to subsequently project these biased directions out of the word embeddings.

A square matrix conceptor C is a regularized identity map, which maps an input to another – in the debiasing domain, a word embedding to its bias [9]. For the exact mathematical definition of a conceptor readers can refer to [10] and [9]. Conceptors can be manipulated through boolean logic. Thus, to project out a bias subspace, one can apply the negated conceptor (representing the bias directions) to the word embeddings. In addition to this, through the use of boolean logic, multiple conceptors generated for various class biases can be combined, enabling joint debiasing [9]. Moreover, a conceptor provides a soft projection [8]. For debiasing this means, that the conceptor dampens the bias directions captured in it. Hence, the soft projection will alter only some components of some embeddings, leaving others largely unaltered [7].

4 Analysis of Religious Bias in Word Embeddings

4.1 Data

Each of the debiasing approaches described is based on different types of data: Conceptor debiasing utilizes a set of unlabeled biased words, Hard debiasing re-

quires equality sets, and SoftWEAT is based on the target and attribute sets of WEAT. This paper will attempt to debias against the religion class, specifically with the subclasses: Christianity, Islam, Judaism. The equality set used for religious multiclass debiasing in Manizini et al.'s paper [11] is extended by hand to include 11 equality sets, which are available for downloading[1]. The attribute sets used in this paper are inspired from Popovic et al.'s work [19].

Finally, the debiasing methods are applied on three established word embedding representations, namely: Word2Vec[2], GloVe[3] and ConceptNet [4].

4.2 Analysis

Social biases are present in the word embeddings when neutral words are more strongly associated with one subclass than another. In this section it is shown what impact these associations have more specifically to each subclass of religion: Christianity, Islam, and Judaism.

In order to quantify captured stereotypes in word embeddings, analogies are scored, as proposed by Bolukbasi et al. [1]. The analogies are then scored via equation (5), where δ is the similarity threshold and $\vec{a}, \vec{b}, \vec{x}, \vec{y}$ are words as given above. The intuition behind this equation is that an analogy capturing relationships well should have directions $\vec{a} - \vec{b}$ and $\vec{x} - \vec{y}$ approach parallelism.

$$S_{(a,b)}(x,y) = \begin{cases} cos(\vec{a} - \vec{b}, \vec{x} - \vec{y}) & \text{if } ||\vec{x} - \vec{y}|| \leq \delta \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Table 1 lists the analogies with a score of over 0.15, that are established within the word2Vec embeddings. As a comparison, the biased analogy established by Bolukbasi et al. [1] and Manzini et al. [11], in addition to some appropriate analogies, are given with their respective scores. Although it follows that the maximal absolute score of equation (5) is 1, in table 1 one can see that established analogies like "*kitten* is to *cat*, as *puppy* is to *dog*", achieve a score of 0.38. Thus, when regarding how high appropriate analogies are scored, biased analogies with an absolute score of higher than 0.15 indicate that these biased analgoies are captured in the word embeddings.

An appropriate analogy concerning religion would be "*Muslim* is to *Islam* as *Christian* is to *Christianity*", which describes the correct correspondence of religion and its members. However, a similarly high classified analogy is "*Christian* is to *judgemental* as *Muslim* is to *terrorist*". This wrong association of religions to terrorist and judgmental is an unjust example of a captured stereotype in the word embedding. The prejudice of Muslims being more strongly associated with violence and terrorism is deeply embedded in society as proven by Sides

[1] https://github.com/thaleaschlender/An-Evaluation-of-Multiclass-Debiasing-Methods-on-Word-Embeddings
[2] https://code.google.com/archive/p/word2vec/
[3] https://nlp.stanford.edu/projects/glove/
[4] http://blog.conceptnet.io/posts/2019/conceptnet-numberbatch-19-08/

and Gross. They hypothesize and confirm that "Americans will stereotype Muslims negatively on the warmth dimension— that is, as threatening, violent, etc" (p.5, [20]).

Table 1: Analogies scoring higher than .15 in Word2Vec

Analogy	score
Appropriate Analogies	
cat is to *kitten* as *dog* is to *puppy*	.38332
Muslim is to *Islam* as *Christian* is to *Christianity*	.27088
Christian is to *Christianity* as *Jew* is to *Judaism*	.26884
Muslim is to *Islam* as *Jew* is to *Judaism*	.24883
Christianity is to *Church* as *Judaism* is to *Synagogue*	.24054
Analogies Exhibiting Stereotypes	
woman is to *homemaker* as *man* is to *programmer*	.26415
Black is to *criminal* as *Caucasian* is to *police*	.07325
Christian is to *judgemental* as *Muslim* is to *terrorist*	.246935
Christian is to *conservative* as *Muslim* is to *terrorist*	.215955
Christian is to *conservative* as *Muslim* is to *liberal*	.177172
Christian is to *judgmental* as *Muslim* is to *uneducated*	.171767
Christian is to *judgmental* as *Muslim* is to *violent*	.171105
Christian is to *greedy* as *Muslim* is to *terrorist*	.166391
Christian is to *judgmental* as *Muslim* is to *liberal*	.155485
Jew is to *hairy* as *Christian* is to *conservative*	.222206
Jew is to *greedy* as *Christian* is to *conservative*	.213083
Jew is to *greedy* as *Christian* is to *judgmental*	.201595
Jew is to *hairy* as *Christian* is to *judgmental*	.197683
Jew is to *liberal* as *Christian* is to *conservative*	.181528
Jew is to *cheap* as *Christian* is to *conservative*	.177668
Jew is to *dirty* as *Christian* is to *conservative*	.176638
Jew is to *familial* as *Christian* is to *conservative*	.173743
Jew is to *hairy* as *Christian* is to *violent*	.168193
Jew is to *dirty* as *Christian* is to *judgmental*	.151427
Muslim is to *terrorist* as *Jew* is to *greedy*	.239060
Muslim is to *terrorist* as *Jew* is to *hairy*	.227352
Muslim is to *violent* as *Jew* is to *greedy*	.207468
Muslim is to *violent* as *Jew* is to *hairy*	.196129
Muslim is to *terrorist*as *Jew* is to *dirty*	.192120
Muslim is to *terrorist* as *Jew* is to *cheap*	.187418
Muslim is to *uneducated* as *Jew* is to *greedy*	.180224
Muslim is to *conservative* as *Jew* is to *greedy*	.172667
Muslim is to *terrorist* as *Jew* is to *familial*	.168889
Muslim is to *liberal* as *Jew* is to *greedy*	.160143
Muslim is to *violent* as *Jew* is to *dirty*	.155248
Muslim is to *conservative* as *Jew* is to *hairy*	.154570

5 Experiments and Results

5.1 Experimental Setup

After the confirmation of religious bias existence two main sets of experiments are held and described below.

The first aims to evaluate the performance of bias removal techniques on a common basis. It does this by observing different quantifications of bias pre- and

post- the application of the debiasing methods. The metrics RNSB, WEAT and MAC are calculated for each word embedding, Word2Vec, GloVe and ConceptNet. We use hard debiasing, Conceptor debiasing with the aperture $\alpha = 10$ and SoftWEAT with $\lambda = 0.5$ and a threshold of 0.5. After each debiasing method, the metrics are calculated anew. Thus, it is possible to evaluate the performance of prior and post debiasing on different word embeddings and debiasing methods in a universal, comparable manner. Since WEAT and MAC are distance measures, the results collected here remain stable over multiple runs. However, to calculate the RNSB metric a logistic classifier is trained on randomly split training and test data. Hence, variability in the RNSB metric is introduced through the individually trained classifier. To counteract this, the RNSB is averaged over 20 runs.

Afterwards, a second set of experiments aims to examine the impact of the SoftWEAT hyperparameters by investigating the impact of hyperparameter λ. This parameter tunes how harshly debiasing is applied and is named as one of the strong advantages of SoftWEAT [19].

5.2 RNSB Metric on Word Embeddings

The results in table 2 show the RNSB values before and after hard debiasing, Conceptor debiasing and SoftWEAT debiasing approaches on word2Vec, GloVe and ConceptNet respectively. The best RNSB scores of each word embedding is highlighted. To statistically analyse whether the RNSB has been improved significantly, a one tailed t-test is performed on all values. The p values are given in table 2 showing that with a significance of $\alpha = 0.05$, it can be concluded that each debiasing method improves the mean RNSB value significantly compared to the non-debiased word embeddings.

Pre-debiasing the word embeddings of ConceptNet carry the least bias, whereas the GloVe word embeddings carry the most bias, according to their RNSB score. Hard debiasing appears to debias the embeddings most efficiently, followed by Conceptor debiasing, whereas SoftWEAT achieves worse results in comparison. This could be attributed to the fact that SoftWEAT only manipulates a collection of words (the identity terminology and its neighbours), whereas the other two debiasing approaches manipulate the whole vocabulary.

The RNSB metric aims to evaluate the bias through a downstream sentiment analysis task. The results show that post debiasing each religion is classified more equally negative with respect to the other religions. Concretely, these improvements for the three debiasing methods on Word2Vec can be seen in figure 1, which depicts the negative sentiment probability for each religion.

The RNSB score decreases as the negative sentiment probability for each religion approaches a sample of the uniform distribution. In figure 1, one can compare each distribution to a fair uniform distribution. Observing this, the non debiased distribution differs from the uniform distribution considerably, whereas the post hard debiasing distribution resembles the uniform distribution the most. This is also indicated by their respective RNSB scores shown in table 2.

Table 2: Relative Negative Sentiment Bias after application of debiasing techniques on Word2Vec, GloVe and ConceptNet

Debiasing Techniques	**Word Embeddings**					
	Word2Vec		***GloVe***		***ConceptNet***	
	RNSB	*p*	*RNSB*	*p*	*RNSB*	*p*
Non-Deb.	0.12339	N/A	0.26033	N/A	0.02276	N/A
Conc. Deb.	0.00682	0.027	0.00024	0.002	0.00775	0.031
Hard Deb.	**0.0**	0.017	**0.00023**	0.002	**0.0**	0.024
SoftWEAT	0.07244	0.032	0.0525	0.002	0.0179	0.035

Furthermore, figure 1 shows that Islam terminology is most likely to be predicted as of negative sentiment. This considerable difference is intuitive when recalling the Muslim and terrorism association captured in the word2Vec embedding, found in the analogies of table 1. It is also interesting to note that after performing Conceptor debiasing, Islam terminology actually becomes the least likely to be predicted of negative sentiment. Thus, Conceptor debiasing has changed the hierarchy of the religions, whereas hard debiasing and SoftWEAT debiasing dampen the original non-debiased distribution.

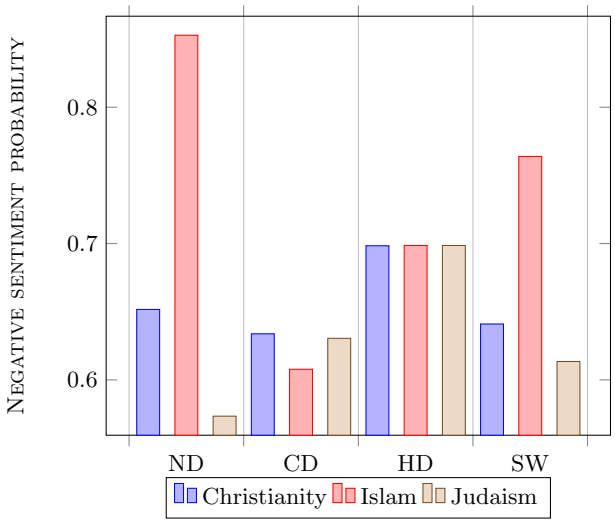


Fig. 1: The negative sentiment probability for Religion terminology from Christianity, Islam and Judaism before and after post processing methods, namely: ND: no debiasing, CD: Conceptor debiasing, HD: hard debiasing and SW: SoftWEAT debiasing

5.3 WEAT and MAC on Word Embeddings

This paper now moves on from the downstream application analysis via RNSB to the geometric analysis of the bias removal methods via WEAT and MAC. Again, to identify the impact of each debiasing method, all values can be compared to the original word embedding prior to any debiasing.

Firstly, the WEAT measurements prior and post the three debiasing methods are shown in table 3. To ease the interpretation of the table, the best scores are bold, whilst scores, which decrease performance to the baseline of the non debiased word embeddings are italic. With the exception of the SoftWEAT application on the ConceptNet embedding, all debiasing methods reduce the WEAT measurements and thus, appear to debias the word embeddings to a given extent.

The performance of the three debiasing techniques in terms of WEAT scores is the same as found within the RNSB evaluation. The hard Debiasing technique performs best, followed by Conceptor debiasing, whereas SoftWEAT's WEAT scores are poor in comparison. In fact, when applying SoftWEAT to ConceptNet, it actually increases the WEAT score, indicating an increase of measured bias. This poor performance could be attributed to the manipulation of less of the embeddings in the vocabulary, as mentioned earlier.

Table 3: WEAT and |1-MAC| after application of debiasing techniques on word2Vec, GloVe and conceptnet - The closer to 0 the better

Debiasing Techniques	**Word Embeddings**					
	WEAT scores			**|1-MAC|**		
	Word2Vec	***GloVe***	***ConceptNet***	***Word2Vec***	***GloVe***	***ConceptNet***
Non-Debiased	0.39469	0.67556	0.76714	0.11787	0.16771	0.00482
Conceptor Debias	0.17112	0.06348	0.30251	**0.00436**	**0.0003**	**0.0030**
Hard Debias	**0.00082**	**0.038215**	**0.00441**	0.11039	0.15603	*0.00624*
SoftWEAT	0.31639	0.40967	*0.83589*	0.07766	0.11871	*0.01367*

In table 3 the MAC scores are presented. In order to ease comparison, the MAC values are subtracted from the optimal value 1. Hence, the closer the MAC values are to 0, the less bias was measured. A similar performance hierarchy of debiasing techniques found in RNSB and WEAT is expected for the MAC scores. Again, to ease comparison, bold and italic fonts are used as described above.

Via the one tailed t-test, the corresponding p values to the MAC scores were calculated. With a significance of $\alpha = 0.01$, the MAC values are all improved compared to their non-debiased version, an exception being both SoftWEAT and hard debiasing when applied to ConceptNet.

Both WEAT and MAC are taken from the notion of measuring bias in cosine distance. The results of both metrics show that the Conceptor debiasing performs well, whilst SoftWEAT performs poorly in comparison. It is interesting to note

that hard debiasing achieves the best RNSB and WEAT scores, yet achieves poor MAC scores - worsening the MAC score within the ConceptNet embeddings. This could be due to the fact that WEAT is a relative measure between two religions and two attribute sets, whereas MAC captures the distance of one religion to all attribute sets. Hard debiasing may introduce new bias by the harsh removal of its religion subspace. This bias introduction may then only be captured in the MAC scores. In fact, when examining the measured mean cosine distance for each religion to each attribute set in word2Vec, one can see that Hard Debiasing improves scores for Judaism, but slightly worsens scores for Christianity and Islam.

In general the results above show that the word embedding ConceptNet carries the least bias as evaluated by MAC and RNSB scores. However, surprisingly, the WEAT score measured in ConceptNet is the worst of all three. The GloVe embeddings seem to carry the most bias concerning the RNSB and MAC metrics, which is intuitive when considering the common crawl data it was trained on.

5.4 SoftWEAT hyperparameter λ experimentation

Having analysed the general performance of all three debiasing techniques above, this paper now turns to the evaluation of SoftWEAT, which has performed most poorly so far. The analysis will examine whether the tuning of the hyperparameter λ may improve the performance within the evaluation metrics used above.

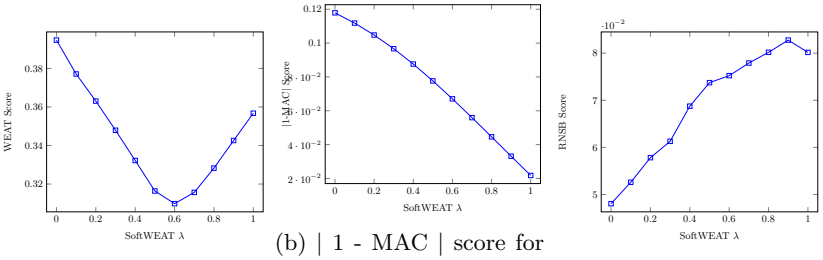
In figure 2a it can be seen that the WEAT score monotonically decreases with increasing values up to a λ of 0.6. From then onwards, the WEAT score steadily increases again. Popovic et a.l [19] report a similar peek in their religious debiasing of Word2Vec. It seems that with a λ higher than 0.6, new bias is introduced by removing one bias too harshly. However, when regarding the |1-MAC| scores in figure 2b, one can see that higher λ values perform better.

When observing the RNSB scores in figure 2c, the tendency that higher λ values lead to a general increase in the RNSB score is shown. One should note, however, that the absolute increase between the values is in the small range of 0.031. The variability of the RNSB framework introduced by its anew training of a classifier at each run in addition to the small range of absolute change in the experiments explains the variability in figure 2c. Figure 2c shows that a good result is already achieved at $\lambda = 0$. This indicates that the RNSB classifications already benefit from the identity terminology of a religion and its neigbours being normalised.

To summarize, it seems that larger λ values improve the bias removal in terms of MAC scores, that a peak value is found in the WEAT scores and that the RNSB scores worsen marginally with higher λs.

6 Conclusion

This paper analysed the debiasing methods of word embeddings via multiple metrics to establish whether a debiasing method could remove religious bias



(a) WEAT score for λ values in the range of [0,1], with a threshold of 0.5.

(b) | 1 - MAC | score for λ values in the range of [0,1], with a threshold of 0.5.

(c) RNSB score for λ values in the range of [0,1], with a threshold of 0.5.

present in the embeddings. For this, this paper has reviewed work showing that social biases persist in word embeddings, whilst briefly showing some possible causes in the data word embeddings are trained on. The investigation of state-of-the-art multiclass debiasing methods is done on Hard debaising, SoftWEAT debiasing and Conceptor Debiasing. This paper evaluates their performance not only on the established WEAT metric but also contributes a performance evaluation on the geometric metric MAC and the downstream metric RNSB. By establishing a common base for the debiasing methods, this paper achieves a more meaningful comparison across methods. To highlight the need of the bias removal, religious bias - as an example of social bias - has been shown to persist in word embeddings by scoring various stereotypical analogies.

It is found that Conceptor Debiasing performs well across all metrics and word embeddings, whereas SoftWEAT, regardless of hyperparameter tuning, performs poorly in comparison. Hard debiasing performs well on RNSB and WEAT scores, however shows shortages when evaluating the removal via MAC - indicating that bias may not be removed as well as previously thought. Hence, to recommend a debiasing technique, which performs well in all bias removal quantifications, Conceptor Debiasing is advised. This comes with the added benefit that this technique is applicable for joint multi-class debiasing and is most flexible in what data it is given to establish its conceptor on.

Finally, this paper calls for more research into establishing a common debiasing approach. Specifically, this approach should perform well in geometric and downstream analysis of bias removal, whilst not decreasing its semantic power. A possible solution could be a combination of a post processing method as investigated in this paper, with a potential pre selection of data to train on to combat implicit bias.

References

1. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: Advances in neural information processing systems. pp. 4349–4357 (2016)

2. Brunet, M.E., Alkalay-Houlihan, C., Anderson, A., Zemel, R.: Understanding the origins of bias in word embeddings. arXiv preprint arXiv:1810.03611 (2018)
3. Caliskan, A., Bryson, J.J., Narayanan, A.: Semantics derived automatically from language corpora contain human-like biases. Science **356**(6334), 183–186 (2017)
4. Garg, N., Schiebinger, L., Jurafsky, D., Zou, J.: Word embeddings quantify 100 years of gender and ethnic stereotypes. Proceedings of the National Academy of Sciences **115**(16), E3635–E3644 (2018)
5. Gonen, H., Goldberg, Y.: Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In: Proceedings of NAACL-HLT (2019)
6. Howard, A., Borenstein, J.: The ugly truth about ourselves and our robot creations: the problem of bias and social inequity. Science and engineering ethics **24**(5), 1521–1536 (2018)
7. Jaeger, H.: Conceptors: an easy introduction. arXiv preprint arXiv:1406.2671 (2014)
8. Jaeger, H.: Controlling recurrent neural networks by conceptors. arXiv preprint arXiv:1403.3369 (2014)
9. Karve, S., Ungar, L., Sedoc, J.: Conceptor debiasing of word representations evaluated on weat. arXiv preprint arXiv:1906.05993 (2019)
10. Liu, T., Ungar, L., Sedoc, J.: Unsupervised post-processing of word vectors via conceptor negation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 6778–6785 (2019)
11. Manzini, T., Lim, Y.C., Tsvetkov, Y., Black, A.W.: Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. arXiv preprint arXiv:1904.04047 (2019)
12. May, C., Wang, A., Bordia, S., Bowman, S.R., Rudinger, R.: On measuring social biases in sentence encoders. arXiv preprint arXiv:1903.10561 (2019)
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
14. Nelson, G.S.: Bias in artificial intelligence. North Carolina medical journal **80**(4), 220–222 (2019)
15. Ntoutsi, E., Fafalios, P., Gadiraju, U., Iosifidis, V., Nejdl, W., Vidal, M.E., Ruggieri, S., Turini, F., Papadopoulos, S., Krasanakis, E., et al.: Bias in data-driven artificial intelligence systems—an introductory survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **10**(3), e1356 (2020)
16. Osoba, O.A., Welser IV, W.: An intelligence in our image: The risks of bias and errors in artificial intelligence. Rand Corporation (2017)
17. Papakyriakopoulos, O., Hegelich, S., Serrano, J.C.M., Marco, F.: Bias in word embeddings. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 446–457 (2020)
18. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
19. Popović, R., Lemmerich, F., Strohmaier, M.: Joint multiclass debiasing of word embeddings. arXiv preprint arXiv:2003.11520 (2020)
20. Sides, J., Gross, K.: Stereotypes of muslims and support for the war on terror. The Journal of Politics **75**(3), 583–598 (2013)
21. Sweeney, C., Najafian, M.: A transparent framework for evaluating unintended demographic bias in word embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1662–1667 (2019)

The fidelity of global surrogates in interpretable Machine Learning

Carel Schwartzenberg[1], Tom van Engers[1], and Yuan Li[2]

[1] University of Amsterdam, Amsterdam, The Netherlands
carel_schwartz@hotmail.com
vanEngers@uva.nl
[2] ING, Amsterdam, The Netherlands
yuan.li@ing.com

Abstract. In this paper, we focus on an interpretable machine learning technique that has increasingly gained attention as of late, named 'Global Surrogates'. When using a global surrogate, an interpretable 'white-box' model is trained on a less-interpretable, but more accurate, 'black-box' model. The original black-box is used to make the predictions, while the white-box surrogate is used to understand the decision making process. A potential problem with Global Surrogates is the fidelity of the white-box surrogate to the original black-box model. To research the fidelity of global surrogates, we perform three experiments. In the first experiment, we find that the Spearman Correlation is the most appropriate metric to measure the fidelity of surrogates. From the results in the second experiment, we find that Logistic Rule Regression (LRR) and RuleFit, two rule ensembles, consistently show high fidelity. Also, we conclude that the fidelity of the different classes of surrogate models depends quite heavily on the type of original black box. Finally, when we look into the fidelity-interpretability trade-off of global surrogates in the third experiment, we conclude that LRR, RuleFit and decision trees perform well in terms of their fidelity-interpretability trade-off.

Keywords: Interpretable Machine Learning · Global Surrogates · Fidelity.

1 Introduction

Currently, increasingly more decisions are being made with the help of Machine Learning models. A lot of focus is on making these decisions with the highest accuracy possible. However, this focus on high accuracy causes algorithms to become increasingly complex [18]. This increase in complexity comes at a price: models are gradually becoming less and less interpretable [5]. Even though theoretically, the calculations that lead to a certain decision are known, it has gradually become more difficult to explain what the exact cause of a certain prediction is.

Multiple stakeholders benefit from insight into the decisions made by Machine learning models. These stakeholders can be categorized into three main groups

[16]: The people using the model to make decisions, the developers of a model and potentially also the human subjects of a model.

The user of a model can benefit from model explanations by gaining new insights into the task the model is used for. Also, relevant explanations can increase the user's trust in the model, allowing the user to rely more confidently on the model. The creators of a model benefit from model interpretability because explanations are a tool to evaluate a model. For example, model interpretability might allow a developer to find that a model makes illogical decisions or that a model is unintentionally biased. Finally, the human subjects of a model might gain from explanations by learning more about the decision being made about them. Additionally, a model that makes decisions about human subjects requires interpretability by law, as defined in the GDPR.

A wide array of techniques has been developed to tackle the challenge of interpreting complex machine learning models [9]. In the research in this paper, the focus is on one such machine learning interpretability method, called 'Global Surrogates'. In Global Surrogates, an interpretable surrogate "white-box" model is trained on the predictions of an existing less-interpretable "black-box" model, in order to interpret the predictions made by the black-box.

Global surrogates have gained popularity in recent research. Surrogate models have been proposed as "well suited to verify a system and detect failures" by interpretability research [16]. Also, the European banking authority [6] includes surrogates as one of their primary examples of interpretability techniques. Furthermore, research by the data science industry sees surrogates as "appropriate for data scientist entrusted with model development" [2].

For an interpretability method to indeed perform well, two main aspects are important. Firstly, the explanations given by the interpretability method need to be understandable and easy to interpret. For global surrogates this is the case; White-boxes, which are used as the surrogates, are defined by their interpretable nature. Secondly, the explanations offered by interpretability methods need to correctly explain the original black-box model: The explanations need to be of high fidelity to the original black-box model. In contrast to the interpretability of surrogates, the fidelity is a potential weakness.

Interestingly, very little research has been done to show that surrogate models produce high fidelity explanations. Thus, multiple questions arise: Are surrogate models indeed appropriate for regulators and data scientists? How high is the fidelity of explanations given by surrogates really? Can surrogates be relied upon to represent complex black-box models? Is using global surrogates worth it, or should a white-box be used instead?

Based on these questions, we perform a set of three experiments. In the first experiment, we research which metric we should use to measure the fidelity of global surrogates. Then, using this metric, we determine the fidelity of multiple classes of global surrogates on multiple classes of black-boxes in the second experiment. Finally, since the higher fidelity surrogates are expected to be more complex and thus less interpretable, we look into the fidelity-interpretability trade-off of global surrogates in the third experiment.

2 Background and related work

Global surrogates are flexible and widely applicable [12]. They are not only model-agnostic, but also the model that is used as surrogate is interchangeable: any interpretable model can be used as the surrogate.

The idea behind surrogate models is borrowed from engineering: Surrogate modelling is concerned with developing and utilizing cheaper-to-run "surrogates" of the original simulation model [14]. In other words, if a simulation is computationally very expensive, a cheaper-to-run surrogate is used to approximate the original simulation.

Research on global surrogates for AI interpretability purposes generally has not received a lot of attention yet. Lakkaraju [11] proposes a novel framework to explain a black-box through decision sets, which are trained to be unambiguous, high-fidelity and interpretable. Ribeiro [15] has done research on explaining complex models through the use of Anchors. In this research, Anchors are if-then rules that sufficiently "anchor" the original prediction locally. On a different note, Bastani [3] looks at the construction of decision trees from a black box. Kuttichira [10] also looks into the approximation of a black-box with a decision tree and proposes a novel way of training the decision tree, making sure to stay close to the original model.

3 Selecting the right fidelity metric

Machine learning theory describes a wide array of metrics that can measure the quality of a model. The available metrics can roughly be divided into two subclasses: categorical metrics and continuous metrics. Categorical metrics measure whether instances have a correct categorical classification, while continuous metrics measure the relatedness of the continuous output of a model to a continuous training label. In general, categorical metrics are used for classification purposes, while continuous metrics are used for regression purposes.

In the context of surrogate classification models, both categorical metrics and continuous metrics can be used. We can measure how many of the categorical classification values of the surrogate match the categorical classifications values of the black-box and thus use a categorical metric. However, we can also measure how far the continuous outputs of the models are apart and thus use a continuous metric.

The question arises which group of methods better reflects how well a surrogate resembles the original model. Or, more specifically: Which metric better reflects if the surrogate model will correctly explain the decisions made by the original model.

Since continuous values capture more information than categorical values, we expect the continuous metrics to perform better than the categorical metrics.

There also exist a variety of continuous metrics. A relevant distinction we can make with continuous metrics is between distance-based continuous metrics and correlation-based continuous metrics. Distance-based metrics measure how

far predictions and labels are apart in the real-valued prediction space, while correlation-based metrics measure how related the predictions and labels are. In the context of fidelity metrics, correlation-based metrics have the advantage of being scale-invariant, which should be a significant advantage when determining the fidelity of surrogates.

3.1 Experiment 1: Fidelity metrics

We will perform an experiment to validate our hypotheses from the previous section and determine which metric is actually best at determining the fidelity of a surrogate.

Datasets Since a single dataset would not be enough to confidently base conclusions on, multiple datasets will be used to evaluate the metrics on. Artificially generated datasets will be used to have a clear grip on the complexity and to be able to generate an infinite variety of datapoints. Initially, data containing relatively few features and simple decision boundaries is used, after which complexity is gradually increased.

Four types of datasets will be generated, each with their own geometrical decision boundary shape. The four geometrical shapes consist of: vertices, Gaussian blobs, circles and moons. The 2D variants of these dataset shapes can be seen in Figure 1.

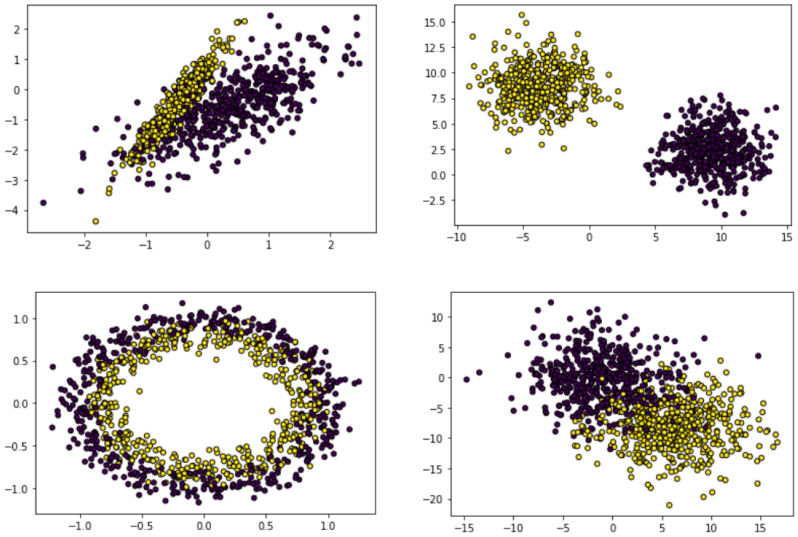


Fig. 1. From left to right, top to bottom: 'Vertices' dataset, 'Blobs' dataset, 'Circles' dataset and 'moons' dataset. The x and y axis represent the feature values, while the color of the points represents the class of the datapoint.

Table 1. The specifications of the generated datasets. The percentage in the last column represent the best average accuracy on the dataset of the models that were trained during this experiment.

Type	# Samples	# Features	# informative	Maximum average classification accuracy
Vertices	1000	2	2	99%
Vertices	1000	2	2	94%
Vertices	1000	10	2	97%
Vertices	1000	10	2	93%
Vertices	1000	10	2	87%
Vertices	1000	10	10	95%
Vertices	1000	10	10	91%
Vertices	1000	20	12	90%
Blobs	1000	2	2	87%
Blobs	1000	10	10	93%
Circles	1000	2	2	92%
Circles	1000	2	2	83%
Circles	1000	2	2	75%
Moons	1000	2	2	90%
Moons	1000	2	2	84%

Multiple key aspects of the generated datasets are changed throughout the experiment, to obtain multiple variations on each dataset type. Aspects that are changed include the number of features and the number of informative features. Also, each of the four types of datasets has specific settings that can be used to tune the class separation and thus to increase or decrease how difficult it is to classify the datapoints. To quantify how challenging each of these datasets is, the 'maximum average classification accuracy' is reported. This is the best average score that the black-box classifiers in the experiment obtained on the dataset.

In the end, a total of 15 dataset configurations is used, as can be seen in Table 1. For each setting of the data generation process, 20 datasets are generated. For each of these datasets, the results are evaluated through 5-fold cross-validation, resulting in a total of 100 training and evaluation cycles per dataset. In the 5-fold cross-validation, the same 4 folds are used for the training of the black-boxes and the white-boxes and the 1 fold is used to apply the fidelity metrics to. A total of 300 datasets is generated (20 per dataset configuration), for each of which we establish which metric agrees on which surrogate is best.

Fidelity measures As fidelity metrics, both categorical and continuous metrics are included, as well as distance- and correlation based-metrics.

Since the focus is on balanced datasets, accuracy suffices as a categorical metric. A second categorical metric that is used, is the Area Under the Curve (AUC). As continuous metrics, both the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) are used. Both these distance-based metrics are included, since it is not clear if squared errors are more impactful than the abso-

Table 2. Overview of the tested fidelity metrics.

Metric	Continuous	Correlation-based
Accuracy	No	No
Area Under the ROC curve (AUC)	Only the predictions	No
Mean Squared Error (MSE)	Yes	No
Mean Absolute Error (MAE)	Yes	No
Spearman Correlation	Yes	Yes
Coefficient of Determination (R^2)	Yes	Yes

lute error in the context of fidelity measurement. The correlation-based metrics of choice are Coefficient of Determination and Spearman correlation. Coefficient of Determination to include a linear correlation measure and Spearman correlation to include a non-linear correlation measure. These six metrics and their characteristics are listed in Table 2.

The accuracy is measured using the output of the original model as labels and the labels produced by the surrogate as predictions. AUC is measured using the binary output of the black box as targets and the continuous output of the surrogate as predictions. The MSE, MAE, Spearman Correlation and Coefficient of Determination are measured using the continuous "probabilistic" output of the original model as labels and the continuous output of the surrogate as predictions. We used these metrics to objectively compare the quality of the surrogate with the original model, and to compare surrogates amongst each others. Which model to choose depends on the specific priorities of the task.

Models Selecting surrogates that are relatively similar in terms of their complexity should lead to more meaningful results. Additionally, surrogate models with both similar and dissimilar inner-working should be included.

For the black-boxes, Random Forests and Neural Networks are selected. The number of estimators for the Random Forest was set to one hundred and Gini was used as the quality criterion. A Neural Network structure with 3 layers was chosen, with a hidden layer width of two times the number of input features. As surrogates, Logistic regression and the rule-ensemble named RuleFit [8] are chosen. The Logistic regression uses SAGA as solver, while RuleFit uses the standard settings as mentioned in the paper by Friedman. The Random Forest and RuleFit are theoretically more related in terms of inner-workings, just like the Neural Network and Logistic regression. Selecting these models leads to a total of four surrogates, two per black box.

Asserting which surrogate is truly best: PMI and ALE To determine which fidelity metric is best at comparing the surrogate to the original model, we need to in some way assert how related the surrogate and original model really are. To do this, we introduce two additional measures that compare the way the surrogate and the original model process the input information into a prediction.

The Permutation Feature Importance (PMI) [7] is a basic and relatively well-known method to measure the importance of each of the features of a machine learning model. PMI will be used to determine how similar the input feature importance's of the surrogate and the original black-box are. The PMI is computed for each of the features of the black box and the surrogate, after which the mean absolute difference is calculated. The result is a single measure that indicates how similar the feature importance's of the surrogate and the original model are.

The Accumulated Local Effect (ALE) [1] plots describe how features influence the predictions made by a machine learning model. The ALE plots of the surrogate and the original black-box will be used to compare the marginal input features effects of the surrogates to those of the original black-boxes. The similarity between the ALE plots of the surrogate and the ALE plots of the black-box is compared through the Mean Squared Error between the ALE measures of both models, resulting in a single measure that indicates the similarity of the marginal feature effects.

The PMI and ALE measures are relatively expensive to calculate and these measures are not as easy to apply to every machine learning model out there. For this reason, the PMI and ALE measures can not simply be used as fidelity measures, but have to be used as guidance to point out which metric should in fact be used as fidelity measure.

3.2 Results of experiment 1

Since there are 15 dataset configurations for which results are gathered, results are too numerous to present for each dataset separately. To solve this, we combined the results into a legible format as follows: For each black-box, two surrogates compete to be the best performing surrogate. Depending on which metric we look at, a different surrogate might seem to perform better. For each of the black-boxes, we determine which metrics agree on which surrogate is best and which do not. The total percentage of times these metrics agree with each other are shown in Table 3. The ALE and PMI metrics give an indication of which surrogate really performs best, i.e. which surrogate is actually closest to the original black-box. Thus, the metrics that frequently "agree" with the ALE and PMI measures can be seen as well-performing measures.

3.3 Conclusion on the fidelity metric

The results show that the absolute difference in relatedness fraction is not very big for most metrics. It should be clarified however that this is partly due to the fact that in some cases, the difference in performance between the two surrogates is significant, while in other cases the difference in performance is small. This means that it is either very clear which surrogate is best and most (if not all) metrics will agree on which surrogate is best, or the difference in performance is very small, resulting in metrics that do not agree on which surrogate

Table 3. The fraction of times the metrics and PMI/ALE agree on which surrogate performs best.

	Acc.	MSE	MAE	R^2	AUC	SpCor
Accuracy (Acc.)	1.00	0.677	0.664	0.677	0.761	0.654
Mean Squared Er. (MSE)	0.677	1.00	0.941	1.000	0.759	0.827
Mean Absolute Error (MAE)	0.664	0.941	1.00	0.941	0.743	0.796
Coefficient of Determination (R^2)	0.677	1.000	0.941	1.00	0.759	0.827
Area Under the Curve (AUC)	0.761	0.759	0.743	0.759	1.00	0.764
Spearman Correlation (SpCor)	0.654	0.827	0.796	0.827	0.764	1.00
Permutation Importance (PMI)	0.648	0.675	0.666	0.675	0.713	**0.720**
Accumulated Locale Effects (ALE)	0.659	0.711	0.723	0.711	0.738	**0.788**

is best. On top of this, for this experiment, "classifying" which surrogate performs best could be seen as a binary classification task: Per black-box, there are only 2 options. This means that a randomly classifying a surrogate as best produces a baseline of 0.5. Combined with the obvious cases, this leads to an easily achievable relatedness score of about 0.6.

As for the best fidelity metric, Spearman Correlation performs best in terms of PMI- and ALE-relatedness. Based on the theoretical considerations at the start of this chapter and the results of the experiment, we thus conclude that Spearman Correlation is the most appropriate metric to use, or at least out of the metrics that were evaluated.

4 Experimental setup

Now that we have found an appropriate fidelity metric, we define a second and third experiment. In the second experiment, we use the Spearman Correlation to measure the fidelity of surrogates. In the third experiment, we look at the fidelity-interpretability trade-off of the surrogates.

4.1 Experiment 2: The fidelity of global surrogates

In the second experiment, we determine how high the fidelity of global surrogates is in general, as well as which classes of surrogates align especially well with certain classes of black-boxes.

Datasets and models To obtain reliable results, we again use a variety of datasets. This time, datasets with real data are used in addition to the generated datasets that were used in experiment 1. This is done to ensure the results are applicable to real-life data and applications.

A range of financial and non-financial datasets is selected for the real datasets, with varying amounts of instances and features, as can be seen in Table 4.

The Mushroom and HELOC datasets are balanced, while the Creditcard, Census Income and Statlog datasets are not. These datasets are balanced artificially through under-sampling of the more available class.

Table 4. The datasets that were used, each with their number of instances, number of features and if the datasets is balanced of itself.

Dataset name	Instances	Features	Balanced
UCI mushroom classification dataset	8124	22	Yes
Home Equity Line of Credit (HELOC)	9871	23	Yes
UCI Creditcard Clients	30000	24	No
UCI Census Income	48842	14	No
UCI Statlog	58000	20	No

Again, 5-fold cross-validation is applied to the datasets to ensure stable results. Just like in the first experiment, the four folds are used as training data for both the black-boxes and white boxes, while the last fold is used to measure the fidelity of the surrogate. The training process will be run 10 times for each dataset.

Multiple black-box and white-box machine learning algorithms are selected to be trained on the datasets. Since there exists a near-infinite number of machine learning models, we limit ourselves to black-box models that are popular in industry. For the white-boxes, well-known models are favoured as well, however, we also include white-boxes that are slightly less well-known, but have shown promising predictive performance in literature.

An overview of the models used can be seen in Table 5. As black boxes, Random Forest, AdaBoost, XGBoost, Neural Networks and SVM are selected. These models are widely used in industry, while this selection also includes a variety of inner-workings. The Neural Network contained 4 layers, with a hidden-layer width of two times the number of features.

As the white-boxes, decision trees, logistics regression and Naïve Bayes are generally well-known and widely used. The rule-ensembles RuleFit and Logistic Rule Regression [17](LRR) are included because previous research suggests these models to have a favourable fidelity-interpretability trade-off. Logistic regression again uses SAGA as solver, the decision trees automatically grid-searches for the optimal tree-depth and for Naïve Bayes the standard settings are used. RuleFit and LRR use the standard settings as suggested in their respective papers.

Table 5. The black-box and white-box surrogates that were tested.

Black-boxes models	White-box surrogates
Random Forest	Decision Tree
AdaBoost Classifier	Logistic Regression
GradientBoost classifier	Naïve Bayes
Support Vector Machine	RuleFit
Neural Network	Logistic Rule Regression

4.2 Experiment 3: Fidelity-interpretability trade-off

Selecting a surrogate often entails more than purely the fidelity of that surrogate: The fidelity-interpretability trade-off of the surrogate can also be important. This is why in addition to evaluating the surrogates purely based on their fidelity, the interpretability of the surrogates will also be quantified. This should give a more complete picture with respect to which surrogate is most fit for a certain black-box and task.

To quantify the interpretability of the surrogates, a methodology from a paper by Molnar is used [13]. Molnar quantifies the interpretability of a machine learning model as a combination of three factors: the 'Main Effect Complexity', the 'Interaction Strengths' and the 'Number of Features'. The 'Main Effect Complexity' represents the complexity of the relationships between the input feature and the prediction, the 'Interactions Strengths' represents the strength of the interactions between the features and the 'Number of Features' represents the number of features that have an influence on the outcomes of the model. By re-scaling and combining these three features, Molnar comes to a single interpretability score for each surrogate, where a score of zero means low relative interpretability and a score of three means high relative interpretability.

The datasets and models used in this experiment are the same real datasets that were used in the second experiment, for the obvious reason that this experiment is meant to give insight into the interpretability of the models that were used in the previous experiment. The generated datasets are not expected to offer additional insights into the interpretability of the models, because of the relative simplicity of these datasets.

5 Results

5.1 Experiment 2: Fidelity of surrogates

For each of the five black-boxes, fidelity results of the five surrogates are gathered over all the datasets. In Figure 2, we report how on how many datasets each surrogate performs best per black-box. Notably, the best performing surrogate varies significantly per black-box, but also depending on the dataset. For Random Forests and XGB, RuleFit is the most frequent best performer. For AdaBoost and SVM, Logistic Rule Regression is the most frequent best performer. For Neural Networks, Logistic Regression performs best.

In Figure 3, the average Spearman correlation over all twenty datasets is reported. The average Spearman Correlations tell a similar story as the results in Figure 2: RuleFit and Logistic Rule Regression perform best in general. This time however, Logistic rule regression performs better than Logistic regression on Neural networks. This is because while Logistic regression performs best on most datasets, it performs poorly on the remaining datasets. A main reason for this is that logistic regression is not able to take on circular decision boundaries, causing it to perform badly on the generated circular datasets as well as on the real datasets that require circle-shaped decision boundaries.

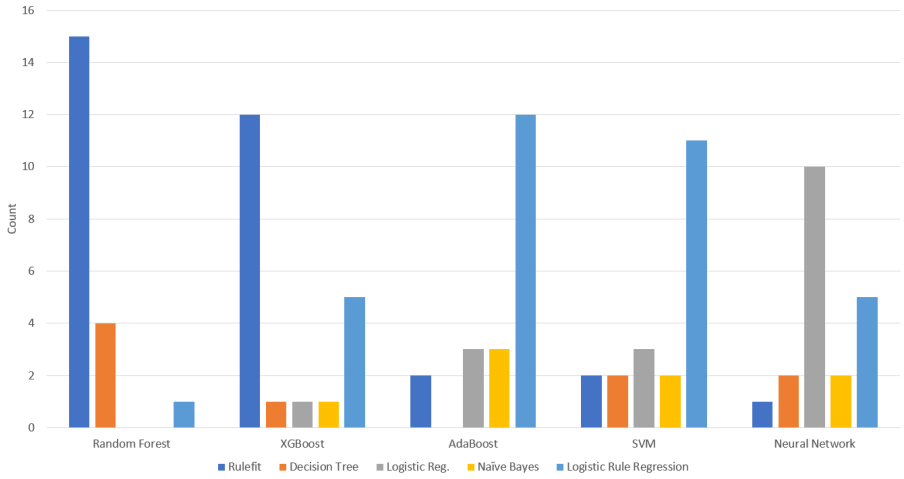


Fig. 2. The frequency a surrogate performs best fidelity-wise on a certain black-box, over all twenty datasets.

5.2 Experiment 3: Interpretability of surrogates

The interpretability experiment has been performed on the exact same models and datasets as the results in the fidelity experiment.

The results can be seen in Figure 4. The interpretability of the surrogates is relatively consistent over the Black-boxes. Overall, decision trees show a high

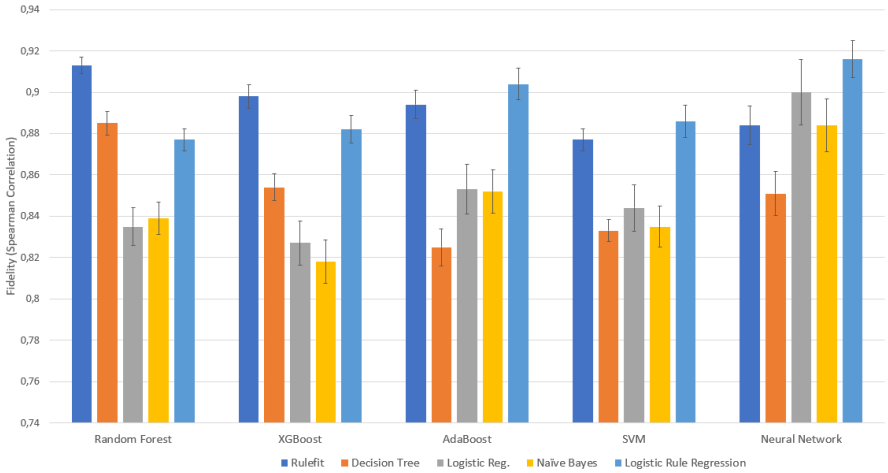


Fig. 3. The fidelity of the five types of surrogates on each of the five types of black-boxes, averaged over all twenty datasets.

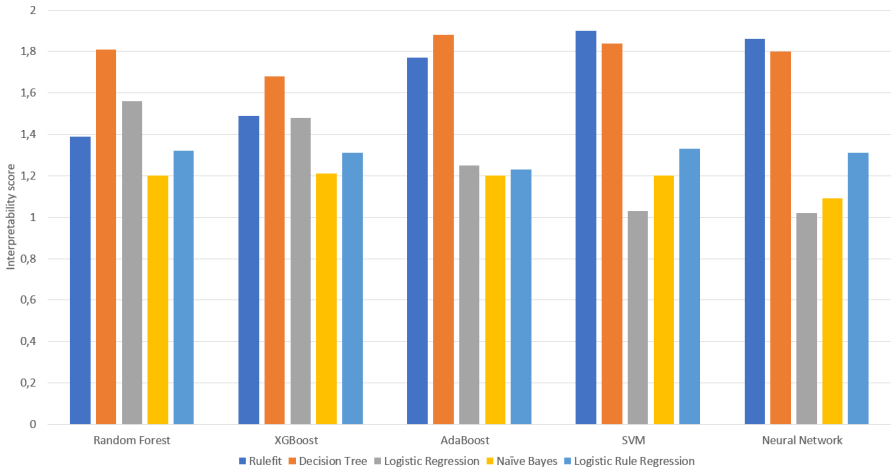


Fig. 4. The interpretability of the five types of surrogates on each of the five types of black-boxes, averaged over the five real datasets. The error bars show the 95% confidence interval.

interpretability score for most black-boxes. RuleFit also performs well, especially on the SVM and Neural Network black-boxes.

If we then combine the fidelity results from experiment 2 and the interpretability results from experiment 3, in both cases using the results of the five real datasets, we produce the fidelity-interpretability plots as shown in Figure 5. For the black-boxes XGBoost, AdaBoost, SVM and Neural Networks, the white-boxes RuleFit and LRR are strong contenders on the Pareto-optimality curve. Decision trees also perform relatively well, especially when focusing on interpretability. For Random Forests, Decision trees trump the other surrogates. It must be noticed that we did not include confidence intervals for the measured interpretability scores. Reason for this is that determining the confidence intervals for Molnar's interpretability score is complicated and exceeds the scope of this paper.

6 Conclusion and future work

In this paper, we investigate how well suited global surrogates are as an AI interpretability method. We performed three experiments: First, we determined which metric is most suitable to measure the fidelity of surrogates. Subsequently, we performed an experiment to determine the fidelity of the surrogates. Finally, an experiment was done to also determine the interpretability of the surrogates.

A variety of white-box surrogate models were trained on a variety of black-boxes, using a multitude of datasets. Based on the theoretical findings and the first experiment, Spearman Correlation appears to be the most appropriate fidelity metric. When we then use the Spearman correlation to measure the fidelity

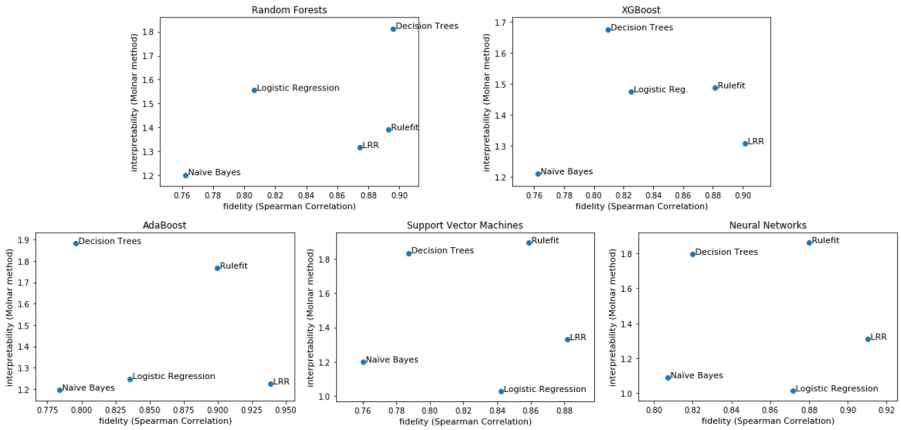


Fig. 5. Fidelity-interpretability trade-off of the surrogates for the five black-boxes. Both the fidelity and interpretability results were averaged over the five real datasets.

of white-box surrogates in experiment 2, we find that the rule-ensembles named Logistic Rule Regression and RuleFit perform well fidelity-wise. The results also show that certain classes of surrogates are better suited for certain classes of black-boxes. The results of the third experiment show that the interpretability of the surrogates is relatively consistent over most black-boxes. In general, Decision trees and RuleFit show high levels of interpretability. If we then plot the fidelity-interpretability trade-off based on the results of experiment 2 and 3, RuleFit, Logistic Rule Regression and Decision trees perform well. Logistic Rule regression does especially well fidelity-wise, while decision trees perform better interpretability-wise.

Back in the introduction, we asked ourselves: “Are surrogate models appropriate for regulators and data scientists?” and “Can surrogates be relied upon to represent complex black-box models?”. We know which surrogates perform best fidelity wise, however, is the level of fidelity of theses surrogates sufficient?

In practice, this is a difficult question to answer. We would say there is not necessarily an a-priori specifiable level of fidelity for a surrogate to be considered reliable, since the required level of fidelity heavily depends on the context in which the interpretability method is applied. For example in healthcare-related applications, surrogate fidelity might be of much higher importance than in more business-related settings, like product recommendations.

We also asked at the start: ”Is using global surrogates worth it, or should a white-box be used instead?”. Most of the time, this comes down to the white-box vs black-box discussion. While black-boxes generally outperform white-boxes in terms of predictive performance, this heavily depends on the dataset and the specific white-box and black-box. Obviously, in cases where a certain white-box performs as well as the best performing black-box, using the white-box is an easy choice. Especially LRR and RuleFit will in some cases perform just as well as

the black-boxes. In most cases, however, the better black-boxes will outperform the white-boxes. In general, this out-performance will entail a few percentage points of accuracy. Selecting the right model for the task here, white-box or black-box, should again be context-dependent: Is a slight increase in accuracy more valuable, or directly available and accurate explanations?

We would thus emphasise that the context in which interpretability is needed, is key. On most datasets and black-boxes, the best performing surrogates reach Spearman Correlation scores of at least 0.9, which translate to AUC scores of at least 0.99. In general, this should lead to fairly correct, high-fidelity explanations.

6.1 Future work

Firstly, the research in this paper specifically focuses on balanced binary classification datasets. However, many real-life datasets do not fit this specification. Therefore, future work includes fidelity research on a wider variety of datasets. Many of the methodologies used in this paper still apply to non-binary, unbalanced and/or regression datasets, however, the related results and conclusions might be different and are therefore worth looking into.

Secondly, the selection of datasets in this paper is focused on data with a relatively low number of features. While many real-life datasets will also contain relatively few features (two dozen at a maximum), black-boxes (and especially deep Neural Networks) are especially well-suited for data with a higher numbers of features. A focus of future work could be to investigate if the conclusions of the research in this paper hold for higher numbers of features. This would also give insight into if global surrogate techniques can be applied to Computer Vision or Natural Language Processing use-cases.

Thirdly, the standard methodology to train a global surrogate is used in this paper: The surrogates are trained directly on the outcomes of the original black-box. Alternative surrogate training methods might however certainly yield better results. Therefore, a second direction of future work could be to investigate ways to increase the fidelity of global surrogates. One such alternative method would be the training strategy used in the ProfWeight algorithm [4]. In ProfWeight, the importance of each training sample is weighted by the performance of the original black-box on that sample, instead of weighting each sample in the training set equally. The ProfWeight paper reports a significant increase in surrogate fidelity using this method.

Fourthly, in the application of global surrogates, it is generally assumed that the same surrogate is used to explain every decision made by the black-box. However, it could be that a certain surrogate shows higher fidelity on some subsections of the data, while it shows lower fidelity on other subsections of the data. Therefore, another potentially interesting direction of future research would be to look into the performance of surrogates on the multitude of subsections of datasets.

Fifthly and finally, since the interpretability of a machine learning model has no clear (mathematical) definition, the literature on the subject has a hard time defining robust interpretability quantification methods. Molnar's interpretability

quantification method, which was used in this paper, is one of the more robust options. The methodology however does have its limitations: It focuses on the functional complexity of a model, instead of the degree of interpretability of a model to a human. Also, Molnar's method doesn't contain a clear way to determine confidence intervals for the estimated level of interpretability of the models. Future work could look into different, potentially more human-focused interpretability quantification methods. Specifically, more focus could be put on the level of interpretability of a variety of knowledge representation formats that can be used to represent the surrogate's decisions. This should lead to a different perspective on the fidelity-interpretability trade-off of global surrogates.

References

1. Apley, D.W.: Visualizing the effects of predictor variables in black box supervised learning models. arXiv: Methodology (2016)
2. Arya, V., Bellamy, R.K.E., Chen, P.Y.: One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. ArXiv **1909.03012** (2019)
3. Bastani, O., Kim, C., Bastani, H.: Interpreting blackbox models via model extraction. ArXiv **1705.08504** (2017)
4. Dhurandhar, A., Shanmugam, K., Luss, R., Olsen, P.A.: Improving simple models with confidence profiles. ArXiv **1807.07506** (2018)
5. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. Communications of the ACM **63**(1), 68–77 (2019). https://doi.org/10.1145/3359786
6. EBA: European banking authority report on big data and advanced analytics (2020)
7. Fisher, A.J., Rudin, C., Dominici, F.: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. **20**, 177:1–177:81 (2019)
8. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles (2008)
9. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An approach to evaluating interpretability of machine learning. CoRR **abs/1806.00069** (2018), http://arxiv.org/abs/1806.00069
10. Kuttichira, D.P., Gupta, S.K., Li, C., Rana, S., Venkatesh, S.: Explaining black-box models using interpretable surrogates. In: PRICAI (2019)
11. Lakkaraju, H., Kamar, E., Caruana, R., Leskovec, J.: Interpretable explorable approximations of black box models. ArXiv **1707.01154** (2017)
12. Molnar, C.: Interpretable Machine Learning, A Guide for Making Black Box Models Explainable (2019), https://christophm.github.io/interpretable-ml-book/
13. Molnar, C., Casalicchio, G.: Quantifying interpretability of arbitrary machine learning models through functional decomposition. ArXiv **1904.03867** (2019)
14. Razavi, S., Tolson, B.A., Burn, D.H.: Review of surrogate modeling in water resources. Water Resources Research **48** (2012)
15. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: AAAI (2018)
16. Ribera, M., Lapedriza, A.: Can we do better explanations? a proposal of user-centered explainable ai. In: IUI Workshops (2019)
17. Wei, D., Dash, S., Gao, T., Günlük, O.: Generalized linear rule models. ArXiv **1906.01761** (2019)
18. Yao, Y., Xiao, Z., Wang, B., Viswanath, B., Zheng, H., Zhao, B.Y.: Complexity vs. performance. Proceedings of the 2017 Internet Measurement Conference (2017)

An Intelligent Tree Planning Approach Using Location-based Social Networks Data

Jan H. van Staalduinen*, Jaco Tetteroo*, Daniela Gawehns, and Mitra Baratchi

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Leiden, The Netherlands
{j.h.van.staalduinen, j.tetteroo}@umail.leidenuniv.nl
{d.gawehns, m.baratchi}@liacs.leidenuniv.nl

Abstract. How do we make sure that all citizens in a city can enjoy the necessary amount of green space? While an increasing part of the world's population lives in urban areas, contact with nature remains important for human well-being. As optional tree planting sites and resources are limited, the best site to plant must be determined. Can we locate these sites based on the popularity of nearby venues? How can we detect groups of people who tend to spend time in tree deprived areas?
Currently, tree location sites are chosen based on criteria from spatial-visual, physical and biological, and functional categories. As these criteria do not give any insights into the number of people benefiting from the tree placement, we propose a new data-driven criterion taking socio-cultural aspects into account. We combine an LBSN mobility data set with a tree location data set, both of New York, as a case study. Using the mobility data we create a venue interaction network from which we extract venue communities. These communities are then scored based on the number of trees in the vicinity of their venues. Applying multi-objective optimization theory, we combine the popularity of venues with the tree density of venue communities to identify locations where planting a tree can benefit the highest number of people and make the largest impact. [1]

Keywords: Urban computing · tree planning · social network analysis · community detection algorithms · mobility data · multi-objective optimization

1 Introduction

As of 2018, 55% of the world's population lives in urban areas, a number which is projected to grow to 68% by 2050 [5]. The North-American continent stands out in particular, where this number is already at 82%. While it is easy to point

* Both authors contributed equally to this project.
[1] This work earlier participated and was selected for the Future Cities Challenge co-organised by Foursquare at NetMob 2019. The work has not been published elsewhere.

out the economical reasons for moving to the city – at least at the first sight [8] – there are certainly downsides attached to urban life. One of them is the inescapable fact that cities, by definition [7], have a higher population density, leading to more built-up areas and thus a scarcer supply of nature than in rural areas. However, as Rohde and Kendle put it, "it is obvious from any casual observation that many human beings do not like to be dissociated from the natural world; as a nation we spend millions of pounds every year on garden and household plants" [21]. Indeed, contact with nature does seem to be linked to human well-being and positive emotional effects and is even said to strengthen urban communities [13, 19]. Apart from socio-cultural benefits, urban greenery can help to mitigate two characteristically urban problems: air pollution due to traffic [14] and (extreme) warmth due to the urban heat island effect [17]. The inclusion of parks and street trees in city landscapes is, therefore, an important aspect of the urban planning process.

To date, socio-cultural arguments play a marginal if not non-existent role in formal frameworks describing criteria for selecting potential tree planting sites. The criteria in these frameworks do not account for the amount of people that are accommodated by the newly planted trees. When following the established criteria, trees may end up in places where they are beneficial to some people, but its effects may not serve the majority of people, or may never reach the people yearning for them most.

To tackle this problem, we propose taking a data-driven approach based on available mobility data which allows considering an additional tree planning criterion. Popular adoption of Location-Based Social Networks (LBSNs) has allowed the collection of valuable data representing the movement of people between venues. Data from the location technology platform Foursquare can be used to construct a network of venues, with users moving between those venues. Priority should be given to sites visited by many people and specifically by people who tend to move between areas lacking trees.

We identify such locations by combining two ways of analyzing the structure of a venue interaction network. By combining the knowledge about (i) venue popularity, and (ii) venue communities with a low tree density, we can detect popular venues within tree deprived communities and thus provide a prioritization that can be used for site selection in the tree planning process, as schematically shown in Figure 1. This prioritization can be embedded within the criteria of established tree planning frameworks that currently lack this socio-cultural value and insight.

Our paper makes the following contributions:

- We describe a novel criterion for potential tree planting site selection based on network communities within a venue interaction network;
- We apply a concept from multi-objective optimization theory to combine this criterion with venue popularity, based on network analysis of venue interaction data from an LBSN;
- We apply this method to prioritize venues as potential tree planting sites in New York City.

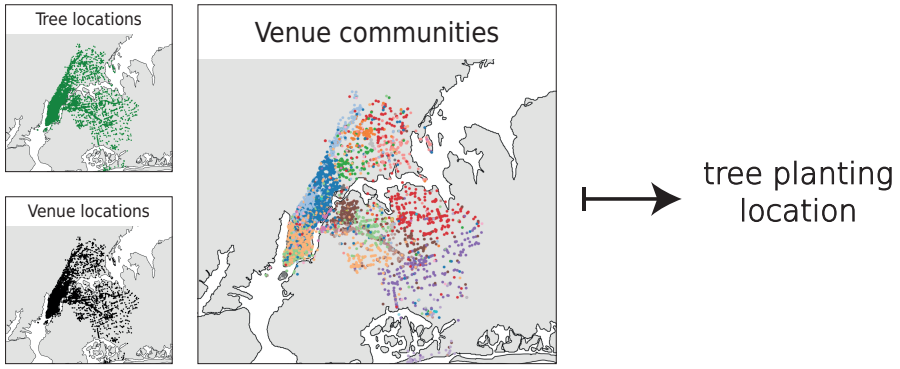


Fig. 1: We combine three types of data (tree locations, venue locations, venue communities) to determine a new criterion which can be used in selecting potential tree planting sites.

The rest of this paper is organized as follows. Section 2 presents the related work. We present our proposed data-driven tree-planning methodology in Section 3. In Section 4 we experiment with the method by implementing it for a specific case in New York City. The results for this are discussed in Section 4.2. Finally, Section 5 presents a number of concluding remarks.

2 Related work

Most of the work in the field of tree planning revolves around selecting appropriate tree species for predetermined planting sites [23, 24]. This reflects the observations by Spellerberg [24] and Pauleit [20] that tree planning is often – or at least has been for some time – an afterthought in the urban design process and characterised by pragmatism. According to an Australian survey, while the visual aesthetic of trees and socio-cultural function of green spaces in the city seem to be important motives for planting trees, the first motive only plays a small role in the tree planning process [22] and the second motive is not reflected in the sparse body of site selection criteria that we could find. The work by Amir and Misgav [2], in which they aim to describe a complete tree planning decision framework, does incorporate criteria on site selection. They define three useful criterion categories, which are *spatial-visual*, *physical and biological* and *functional*. Criteria relating to the socio-cultural function of green spaces, however, are missing. We observed several works describing site selection criteria [10, 20], but those fall within the category of *physical and biological* criteria that are essential for the survival of the tree. Moriani [14] did use population density in their planting priority index, but as they focused on the air pollution-reducing quality of trees, this still falls within the category of *functional* criteria. We believe then, that the body of site selection criteria is still incomplete and that we can contribute to this framework

by introducing a new socio-cultural criterion which takes people movement into account.

As a way to capture the general movement patterns of people within cities, we utilize data collected by LBSNs. As defined by Zheng [25], social networks are social structures that consist of individuals connected to each other via specific types of interdependencies. In LBSNs these individuals are connected through their shared experience interacting with the locations in the network. Oftentimes, in LBSNs users announce their visit to venues through a so-called check-in option. The check-in data can provide information about the movement of people between a network of venues. The structure of such a network can be explored to find underlying patterns. For instance, locations can be grouped based on the similarity between user profiles [12]. Hung et al. [9] use these user profile similarities to find user communities. Girvan and Newman [6], however, use clustering algorithms on the full network to detect communities, eliminating the need for individual trajectories. Noulas et al. in [18] has studied the spatial network of venues derived from such data and proposed a variant of gravity mobility models using inter-venue connectivity information. Most of these approaches have considered studying the network properties of LBSN data without considering how such information can be used in improving urban aspects. Recently, Arp et al. [3] have shown how such data can be used in optimising the state of traffic within the city. In this paper, we aim to study whether such data can be used for improving decision making regarding the optimal allocation of resources, notably in this case the green space, throughout the city.

3 Methods

In this section, we introduce our proposed method. First, we describe two separate possible indicators and how they can be used to define objectives for planting trees (Sections 3.1 and 3.2). Then, we argue that the best way to use them is by combining them using multi-objective optimization theory (Section 3.3), thereby forming the method we propose in this paper.

3.1 Venue popularity

A first possible approach to maximize the impact of planting a tree, is to plant it near a place where many people pass by. From this perspective, the goal is to find the venue that is maximally popular among visitors. To find this place we compute the degree of all nodes in the undirected network graph $G = (V, E, W)$, where nodes $v \in V$ are venues and edges $e = (v_1, v_2), e \in E$ movements of people between two venues v_1 and v_2, with weight $w_e \in W$ as the number of movements between the pair of venues. The degree of a node v is then defined as the sum of the weights of the edges that are connected to it:

$$\deg(v) = \sum_{e \in \{(u,v) | u \in \mathrm{adj}(v)\}} w_e \tag{1}$$

3.2 Venue community tree density

Although trees near popular venues may reach many people, we will still be missing those who visit other venues. Naively, one might say that an additional, or: parallel, objective could be to then look for venues that have the least trees in the vicinity. This approach would, however, discard the reality that people move about and that people are thus prone to visit multiple venues. A single venue that has few trees in its vicinity might not be a major problem if the usual crowd for this venue also regularly visits other venues that do have more trees in the neighbourhood. Using LBSNs, we can actually use this observation in our objective. To this end, we introduce a measure we call the *tree density coefficient*. This measure intends to highlight *groups* of *related* venues that have a low tree density, instead of *single* venues that have a low tree density. A relation between venues, in this sense, is determined by people travelling often between those venues.

Using graph theory parlance, these related venues can be discovered through the task called *community detection*. A community is a group of nodes of which the nodes are densely connected with each other, but much less with the rest of the network [6]. To detect the communities, we use the Louvain community detection algorithm [4]: a fast algorithm that is able to find communities with high quality. The algorithm performs based on the optimization of modularity, a measure that compares the density of connections within a community with the density between communities. Modularity, as defined by Newman et al. [16], is computed as in Equation 2:

$$Q = \frac{1}{2m} \sum_{ij} \Big[A_{ij} - \frac{k_i k_j}{2m} \Big] \delta(c_i, c_j) \tag{2}$$

Here, A_{ij} is the adjacency matrix holding the number of edges between nodes i and j, m the number of edges in the network, k_i the degree of node i and $\delta(c_i, c_j)$ a delta function that returns 1 if i and j are assigned to the same community and 0 otherwise.

As it is computationally heavy to compute the modularity of a community, the Louvain algorithm uses heuristics to approximate it. Therefore, it does not necessarily return the best community layout. To gain confidence in the robustness of our communities, we choose to run the algorithm 1,000 times in our own experiments, to create a large number of community layouts.

To compute the tree density coefficient for a venue, we first count the number of trees in the vicinity of the venues. We approximate this vicinity by creating a grid of the city, thereby discretizing the geographic space into grid cells, where each grid cell is 50 by 50 meters, calculated using Universal Transverse Mercator coordinate system [11]. Each venue v_i is mapped to a cell in the grid and is assigned the number of trees in the cell as its *venue tree density* vtd_i.

We compute the *community tree density* ctd_i for a venue v_i by averaging the vtd_i with the venue tree densities of all the other venues in its community C_i, over multiple iterations k of the community detection algorithm:

$$\mathrm{ctd}_i^k = \frac{1}{|C_i|} \sum_{v_j \in C_i} \mathrm{vtd}_j, \quad 0 < k \leq k_{\max} \tag{3}$$

In the end, the *tree density coefficient* c_i for a venue v_i is its average community tree density value over all iterations of the community detection algorithm:

$$c_i = \frac{1}{k_{\max}} \sum_{t=1}^{k_{\max}} \mathrm{ctd}_i^k \tag{4}$$

3.3 Joining both objectives through multi-objective optimization

The two objectives discussed above, venue popularity and venue tree density, can both be important in discovering the most suitable location(s) for one or more new trees. Indeed, a venue with a low tree density coefficient could have only one visitor, whereas other venues in the same community that have a similarly low tree density coefficient could have many visitors. In this case, the latter venue(s) would be more appropriate as a tree planting site. It is therefore important to take both objectives into account. To achieve this, we borrow a method from multi-objective optimization theory, namely the Pareto front.

We join the venue degrees, i.e. the popularity of venues, with community-based tree density coefficients by detecting the set of venues that are Pareto efficient, i.e., the venues that are found by minimizing the tree density coefficient and maximizing the influence of the venue: the optimal trade-offs between the two measures. Also called the Pareto frontier, the venues in this set meet our criterion of helping most people needing trees. Tree planners could choose any of the venues along the Pareto frontier, depending on their preference towards either of the two measures.

4 Experiments

4.1 Data sources

City of choice: New York We conducted a case study to investigate the implementation and workings of our criterion using real data. For this, we chose to focus on New York City as data on both venue interactions and tree locations were richly available.

We used two data sets to construct our criterion. We used venue interaction data of New York, provided by Foursquare as part of the Future Cities Challenge 2019, to create the venue interaction network. To assign tree density scores, we used a Street Tree Census data set [1]. In the remainder of this section we describe the properties of the venue interaction data and street tree data set, respectively, and how we processed them to implement our methods.

Table 1: Description of New York data set (Foursquare).

	Original	**After pre-processing**
no. of venues (nodes)	17,975	15,803
no. of interactions (edges)	7,919,999 (directed, parallel)	248,597 (undirected)

Venue interaction data Foursquare City Guide is a mobile app that recommends places to its users based on their likes or check-ins. The Foursquare venue interaction data set comprises of two parts: venues and movements between them. Venues in this set are locations people can visit. Venue coordinates are recorded, as well as their name and a category. Movements are recorded when individuals make consecutive check-ins at different locations.

The data set contains information on ten different cities around the world. As we focused on New York in this case study, we used the New York data, but it should be noted this study is applicable to any of the other nine cities, provided we have access to a corresponding tree location data set. The data was collected between April 2017 and March 2019.

As not all venues found in the movement data occur in the venue information data, we considered only the venues with known locations for the construction of the network. Additionally, we observed that some venues were only connected within small subgraphs, 'connected components', of less than 3 venues and did not have any edges to the large, main connected component in the graph. These 86 venues were omitted. In the end, we were able to use 15,803 of the 17,975 venues in our analysis. We used this data to create a network where nodes were represent venues and the edges represent movements between them. We combined the many parallel interactions between venues into singular weighted edges between the venues, where the edge weight denotes the number of interactions between two given venues. Later, we used this data as input for both the detection of nodes with high node degrees (see Section 3.1) as well as the Louvain algorithm (see 3.2). In Table 1, we provide a comparison between the original data set and the pre-processed data set.

Street Tree Census The Tree Census data set contains information on street trees in New York City and surrounding cities. It contains information on among others the *species* and *health* of the trees, as well as their *longitude* and *latitude*. As only street trees were counted, trees in parks were not taken into account in the tree survey and are therefore not present in the data set.

As discussed in Section 3.2, we discretized the geographic space into a grid, counting the number of trees per cell to obtain a measure for the tree density around the location of each venue. To provide insight into the data, we show the tree counts over grid cells in Figure 2.

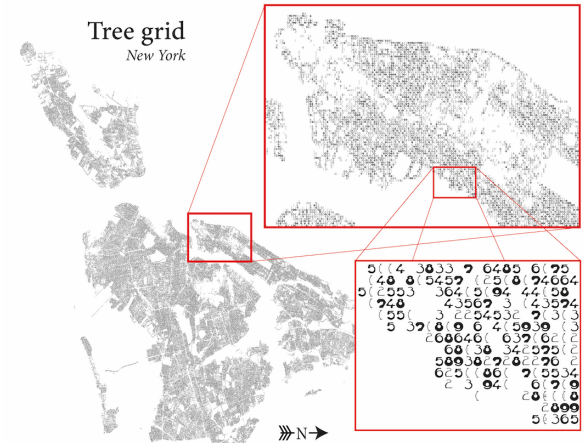
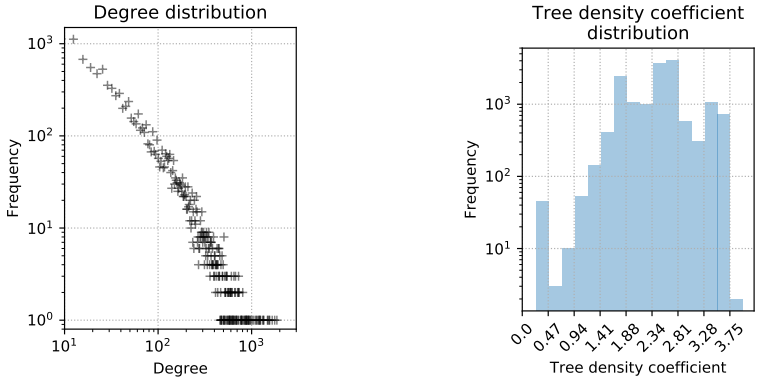


Fig. 2: The trees within part of the New York street tree dataset, discretized into grid cells of 50 by 50 meters. Shown as a heatmap-like data plot using 'FatFonts' [15].

4.2 Results

Venue popularity We computed the venue popularity as the degree of each node and observed that the distribution follows a power law (see Figure 3a), as is generally the case in scale-free networks modeling natural phenomena. To decide which venues would be interesting as a tree planting site according to this method, one should prioritize venues with higher degrees.



(a) (b)

Fig. 3: The power law distribution of venue degrees (a) and distribution of tree density coefficients (b).

Venue community tree density We used the Louvain community detection algorithm as implemented in the Python `NetworkX` package. We set the resolution to 0.5 to find decently small communities. One such community lay out is shown in Figure 5a.

As the communities are detected using the heuristic Louvain algorithm, we averaged the community tree density of the venues over 1,000 runs of the algorithm, each time possibly detecting slightly different communities in the network, to obtain their tree density coefficients.

To find tree-deprived communities, we combined the locations of the venues within the communities with the tree locations in the street tree data set. First, we calculated the tree density for each venue. Then, the average tree density of the venues in the community was computed and returned to each of those venues as its community tree density.

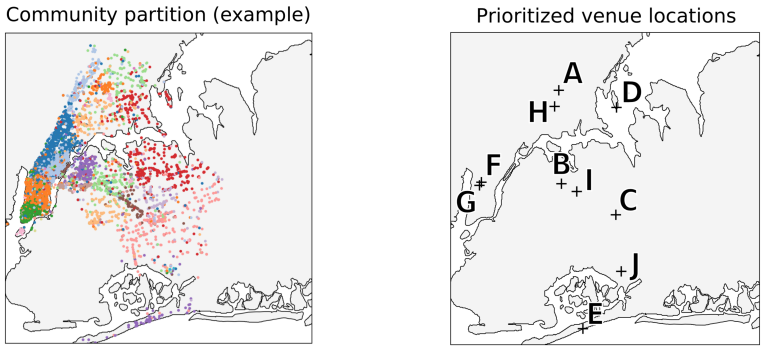
We show the distribution of the tree density coefficient values in Figure 3b. The distribution is slightly skewed to the right, which means most communities are filled with trees. Some, however, would still benefit from planting more. Prioritization for tree planting sites using this method should be given to the venues with the lowest coefficients.

Fig. 4: The distribution of venues according to degree and tree density coefficient. The Pareto frontier shows the venues with the optimal tree planting location according to our criterion. Venue labels correspond with Figure 5b and Table 2.

Joining both objectives through multi-objective optimization To select the most impactful planting locations, we combined both measures. This results in the distribution of venues and associated Pareto frontier as shown in Figure 4. Here we minimize the tree density coefficient of the venues while maximizing their degree. These venues are highlighted by the Pareto frontier and should be

prioritized according to our new criterion. To indicate the locations of the venues on the Pareto frontier, we show the venues on a map in Figure 5b and provide additional insights into the data in Table 2.

It is noteworthy that one of the selected venues (venue H) is a rose garden, amidst a park lush with trees. This is explained by the fact that the tree data set contains only street trees and no park trees. Additionally, we found upon inspection using Google Street View that some of the venues (most notably venues A, B, D, G and H) do seem to be near a considerate amount of trees. When inspecting these locations in the tree data base[2], we see that there are either only a few (venues B and G) or no trees (venues A, D, E and H) recorded in the immediate vicinity of the venues. We see that along with park trees, trees on private grounds are also not recorded.



(a) One of the 1,000 community partitions.

(b) Optimal tree planting locations (see Table 2).

Fig. 5: Map of New York City showing the optimal tree planting locations based on community structures.

5 Conclusion

In this paper, we propose a novel criterion that can be used when selecting potential tree planting sites. The nature of the criterion is socio-cultural, capturing people movement between venues and tree-lacking (social) communities into one measure. Having implemented the measure for a case study on New York City, we show that the measure is applicable in the field and can be used to support decision-makers by providing them with optional planting sites along a Pareto frontier.

[2] The tree database can be explored on a map at `https://tree-map.nycgovparks.org/`, last visited 9 September 2020.

We want to note that our approach depends heavily on the quality of the available data. Regarding the tree data, we see that some venues indicated by our criterion as tree lacking seem to actually be in a green area. We believe that the application of our method can be improved with a more detailed tree location data set. Then, the criterion proposed in this paper can be a meaningful addition to the established site selection criteria.

Regarding the venue communities, we are aware that the used data set includes only venues selected and listed by Foursquare. Amongst those venues are major train stations, schools and other public buildings. The movements between the venues and hence also the venue communities used to find optimal planting locations only represent people that are using Foursquare, other inhabitants are not represented in the data. Unfortunately, full movement data is almost always proprietary. We would like to mention that the venue network could also be estimated based on other, more representative data.

We conclude that the newly introduced socio-cultural approach to finding a tree planting site that benefits different communities of city dwellers is feasible and can be easily implemented by urban planning organizations. Integration of this approach depends on the availability of detailed records of existing trees and movement data of city inhabitants.

Acknowledgments

We thank Foursquare and the organisation of Netmob for organising the Future Cities Challenge and providing us with access to Foursquare's venue interaction data set.

Table 2: The venues that are, according to the Pareto analysis, the most efficient to place trees next to.

Degree	Tree density coefficient	Venue ID	Venue Name	Latitude	Longitude	Venue Category
A 1864	2.09323399	4b637f59f964a5207b7e2ae3	MTA Subway - West Farms Square/E Tremont Av (2/5)	40.8402	-73.8800	Metro Stations
B 1561	1.94936904	4f940fe7e4b059d7da88be53	Junction Blvd	40.7491	-73.8694	Miscellaneous Shops
C 1212	1.86431926	4e7647cffa7605970163202l	MTA Subway - 179th St (F)	40.7125	-73.7846	Metro Stations
D 853	1.70625431	4bace08af964a520cf143be3	Sammy's Fish Box Restaurant	40.8390	-73.7836	Seafood Restaurants
E 532	1.55978734	4cc86db294e1a0933e6c978b	Rockaway Beach - 116th Street	40.5779	-73.8359	Beaches
F 305	1.18353191	4abcfe4bf964a520fa8720e3	Hulu Theater	40.7509	-73.9941	Music Venues
G 112	1.18192331	4c516433d2a7c9b6c4c61911	Bean & Bean Organic Coffee	40.7472	-73.9971	Coffee Shops
H 98	0.87556379	4debdb6b52b11677f060802e	Peggy Rockefeller Rose Garden	40.8592	-73.8735	Gardens
I 40	0.41121454	4d93a4489ef2721e6bffc3d2	I-495 / Grand Central Parkway Interchange	40.7400	-73.8455	Intersections
J 12	0.39198899	4e26fd0f1f6eb1ae139ad929	TSA Security Screening	40.6457	-73.7762	General Travel

Bibliography

[1] NYC Parks Recreation - TreesCount! 2015 Street Tree Census. `https://data.cityofnewyork.us/Environment/2015-Street-Tree-Census-Tree-Data/uvpi-gqnh`. Accessed: 9 September 2020.

[2] Shaul Amir and Ayala Misgav. A framework for street tree planning in urban areas in israel. *Landscape and Urban Planning*, 19(3):203–212, 1990.

[3] Laurens Arp, Dyon van Vreumingen, Daniela Gawehns, and Mitra Baratchi. Dynamic macro scale traffic flow optimisation using crowd-sourced urban movement data. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 168–177, 2020.

[4] Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics Theory and Experiment*, 2008, 04 2008.

[5] UN DESA. World urbanization prospects, 2018.

[6] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[7] Susan A Hall, Jay S Kaufman, and Thomas C Ricketts. Defining urban and rural areas in us epidemiologic studies. *Journal of urban health*, 83(2):162–175, 2006.

[8] John R Harris and Michael P Todaro. Migration, unemployment and development: a two-sector analysis. *The American economic review*, pages 126–142, 1970.

[9] Chih-Chieh Hung, Chih-Wen Chang, and Wen-Chih Peng. Mining trajectory profiles for discovering user communities. In *Proceedings of the 2009 International Workshop on Location Based Social Networks*, LBSN '09, page 1–8, New York, NY, USA, 2009. Association for Computing Machinery.

[10] Chi Yung Jim. A planning strategy to augment the diversity and biomass of roadside trees in urban hong kong. *Landscape and Urban Planning*, 44(1):13–32, 1999.

[11] Richard B Langley. The utm grid system. *GPS world*, 9(2):46–50, 1998.

[12] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, New York, NY, USA, 2008. Association for Computing Machinery.

[13] Cecily Maller, Mardie Townsend, Anita Pryor, Peter Brown, and Lawrence St Leger. Healthy nature healthy people:'contact with nature'as an upstream health promotion intervention for populations. *Health promotion international*, 21(1):45–54, 2006.

[14] Arianna Morani, David J Nowak, Satoshi Hirabayashi, and Carlo Calfapietra. How to select the best tree planting locations to enhance air pollution removal

in the milliontreesnyc initiative. *Environmental Pollution*, 159(5):1040–1047, 2011.

[15] Miguel Nacenta, Uta Hinrichs, and Sheelagh Carpendale. Fatfonts: combining the symbolic and visual aspects of numbers. In *Proceedings of the international working conference on advanced visual interfaces*, pages 407–414, 2012.

[16] Mark EJ Newman. Analysis of weighted networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 70 5 Pt 2:056131, 2004.

[17] Briony A Norton, Andrew M Coutts, Stephen J Livesley, Richard J Harris, Annie M Hunter, and Nicholas SG Williams. Planning for cooler cities: A framework to prioritise green infrastructure to mitigate high temperatures in urban landscapes. *Landscape and Urban Planning*, 134:127–138, 2015.

[18] Anastasios Noulas, Blake Shaw, Renaud Lambiotte, and Cecilia Mascolo. Topological properties and temporal dynamics of place networks in urban environments. In *Proceedings of the 24th International Conference on World Wide Web*, pages 431–441, 2015.

[19] William LI Parry-Jones. Natural landscape, psychological well-being and mental health. *Landscape research*, 15(2):7–11, 1990.

[20] Stephan Pauleit. Urban street tree plantings: identifying the key requirements. In *Proceedings of the Institution of Civil Engineers-Municipal Engineer*, volume 156, pages 43–50. Thomas Telford Ltd, 2003.

[21] CLE Rohde and AD Kendle. Human well-being, natural landscapes and wildlife in urban areas a review. 1994.

[22] Sudipto Roy, Aidan Davison, and Johan Östberg. Pragmatic factors outweigh ecosystem service goals in street tree selection and planting in south-east queensland cities. *Urban Forestry & Urban Greening*, 21:166–174, 2017.

[23] Henrik Sjöman and Anders Busse Nielsen. Selecting trees for urban paved sites in scandinavia. *Urban Forestry & Urban Greening*, 9(4):281–293, 2010.

[24] Ian F Spellerberg and David R Given. Trees in urban and city environments: a review of the selection criteria with particular reference to nature conservation in new zealand cities. *Landscape review*, 12(2):19–31, 2008.

[25] Yu Zheng and Xiaofang Zhou. *Computing with Spatial Trajectories*. Springer Publishing Company, Incorporated, 1st edition, 2011.

Gaining Insight into Determinants of Physical Activity using Bayesian Network Learning

Simone C.M.W Tummers[1], Arjen Hommersom[2,3], Lilian Lechner[1], Catherine Bolman[1], and Roger Bemelmans[4]

[1] Faculty of Psychology, Open University of the Netherlands, Heerlen, The Netherlands
[2] Faculty of Science, Open University of the Netherlands, Heerlen, The Netherlands
[3] Department of Computer Science, Radboud University, Nijmegen, The Netherlands
[4] Zuyd University of Applied Sciences, Heerlen, The Netherlands

Abstract. Bayesian network modelling is applied to health psychology data in order to obtain more insight into the determinants of physical activity. This preliminary study discusses some challenges to apply general machine learning methods to this application domain, and Bayesian networks in particular. We investigate suitable methods for dealing with missing data, and determine which method obtains good results in terms of fitting the data. Furthermore, we present the learnt Bayesian network model for this e-health intervention case study, and conclusions are drawn about determinants of physical activity behaviour change and how the intervention affects physical activity behaviour and its determinants. We also evaluate the contributions of Bayesian network analysis compared to traditional statistical analyses in this field. Finally, possible extensions on the performed analyses are proposed.

Keywords: Machine Learning · Bayesian Network · E-health Intervention · Structure Learning · Physical Activity

1 Introduction

Nowadays there are various e-health intervention platforms that employ integrated behaviour change techniques in order to change health-related-behaviour of participants, for example increasing physical activity. These interventions apply theoretical psychological methods to influence behavioural determinants, which are factors determining a certain behaviour. These general techniques are translated to behaviour change strategies by tailoring the theoretical method to the target population and intervention setting [1]. To measure the effects of such interventions, various research studies have been performed, assessing physical activity with tools such as questionnaires and activity trackers. While there is now a good understanding of what the most important determinants for increasing physical activity are, little is known about how these determinants interact. Improved understanding of these relationships could be used to improve existing e-health interventions.

Supervised machine learning techniques are used to identify relationships underlying data with labeled input and output, and predict output results for a given input. These techniques could for example be used to model relations between diseases and symptoms and give expectations about the presence of various diseases given symptoms. Bayesian networks [8] represent probabilistic relationships between a set of variables, where relationships between the input variables can also be investigated. Such networks can make probabilistic predictions and provide a visual insight in relations among all variables of interest, thereby providing a potential useful tool to better understand determinants of physical activity.

In this article, a Bayesian network model is learned from data from a single intervention study, i.e., the *Active Plus intervention* [12], aiming at influencing physical activity behaviour among older adults. We discuss ways to learn from these complex data containing a significant amount of missing values. Based on these initial findings, results from previous analyses are compared to results from applying the Bayesian network model to the same data, to examine the added value of this technique compared to traditional ones. We show that learning a Bayesian network model for measurement data from the Active Plus project indeed reveals conditional dependence and independence relations that provide new insights and explanations for previously found results.

This paper is organised as follows. Section 2 provides technical background about methods and algorithms. Section 3 provides a description of the data and intervention study at hand, and how the data has been pre-processed. Furthermore, the analysis based on the Bayesian network model is explained including a description of the applied learning strategy, and a missing data analysis to select appropriate methods for handling the missing data. Then, in Section 4, results are given about the comparison of evaluated methods, and the comparison of the results from the Bayesian network model, determined using the best method, and the results from previous analyses. Finally, Section 5 concludes this paper and elaborates on possible extensions.

2 Preliminaries

This section gives an overview of the theoretical background relevant to perform the case study analyses, including a brief introduction of the modelling approach.

2.1 Bayesian network model

A Bayesian network [8] is a probabilistic graphical model represented as a directed acyclic graph $G = (V, E)$, where the set of nodes V represent random variables, and the set of arcs E represent probabilistic independencies among the variables. Associated with each node is a conditional probability distribution of that variable given its parents. The graphical structure implies conditional independence statements. Let $V = \{X_1, \ldots, X_n\}$ be an enumeration of the nodes in a Bayesian network such that each node appears after its children,

and let Π_i be the set of parents of a node X_i. The local Markov property in the Bayesian network states that X_i is conditionally independent of all variables in $\{X_1, X_2, \ldots, X_{i-1}\}$ given Π_i for all $i \in \{1, \ldots, n\}$. These local independences imply conditional independence statements over arbitrary sets of variables.

The joint probability distribution over discrete variables follows from the conditional independence propositions and conditional probabilities:

$$\mathbb{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i \mid X_1, \ldots, X_{i-1}) = \prod_{i=1}^{n} \mathbb{P}(X_i \mid \Pi_i),$$

where the first equation follows from the usual chain rule in probability theory and the second from the local Markov property. Note that the conditional probabilities $\mathbb{P}(X_i \mid \Pi_i)$ correspond to the arcs in the Bayesian network specification. In continuous Bayesian networks, usually a linear Gaussian distribution is assumed, where the joint density is factorised where each $X_i \mid \Pi_i \sim \mathcal{N}(\beta\Pi_i+\alpha, \sigma^2)$.

A temporal Bayesian network is an extension to the static counterpart in that it is a Bayesian network model over time, where the nodes represent the random variables occurring at particular time slices. The temporal Bayesian network model is subject to the condition that arcs directed to variables in previous time slices cannot occur. In case the temporal Bayesian network is time-homogeneous (or time-invariant), these models are also called dynamic Bayesian networks [6]. Since in this case study there are only a few time slices and differences between these slices are not constant, we do not assume time-invariance in the remainder of this paper.

2.2 Learning Bayesian networks

The following three common classes of algorithms are used to learn the structure of Bayesian networks from the data: constraint-based algorithms which employ conditional independence tests to learn the dependence structure of the data, score-based algorithms which use search algorithms to find a graph that maximises a goodness-of-fit scores as objective function, and hybrid algorithms which combine both approaches. Recent research has shown that constraint-based algorithms are often less accurate and seldom faster and hybrid algorithms are neither faster nor more accurate [11]. For this reason, we focus in the remainder of this paper on score-based structure learning algorithms, where local search methods are used to explore the space of directed acyclic graphs by single-arc addition, removal and reversal. In particular, we apply tabu search to the physical activity data in this case study as empirical evidence shows that this search method typically performs well for learning Bayesian networks [5, chapter 13.7].

There are several model selection criteria that are used in the search-based structure learning algorithms, where in this paper we have chosen the commonly-used Bayesian Information Criterion (BIC) [9]. To fit the parameters we have chosen a uniform prior distribution over the model parameters [4].

Algorithm 1 Structural EM algorithm, given $(M_0, \mathbf{o})$:

for $n = 0, 1, \ldots$ until convergence or predefined maximum number of iterations reached **do**
  Compute Θ^{M_n} using a parameter learning algorithm.
  Expectation-step:
  compute $\mathbf{h}^* = \arg\max_{\mathbf{h}} \mathbb{P}(\mathbf{h} \mid \mathbf{o}, M_n)$
  Maximization-step: apply structure learning to determine M_n using data $\mathbf{h}^* \cup \mathbf{o}$
  if $M_n = M_{n+1}$ or if stopping criterion is met **then**
    return M_n
  end if
end for

2.3 Handling missing data

Learning Bayesian networks with missing data is significantly harder as the log-likelihood does not admit a closed-form solution if values are missing. In this paper, we assume that data are missing at random, for which commonly used methods are listwise deletion, pair-wise deletion, single imputation, multiple imputation [7]. The deletion approaches omit (observed) values from analyses. In the listwise deletion approach on the one hand, all observations with missing values at any measurement are omitted completely. On the other hand, the pair-wise deletion method does not require complete data on all variables in the model, and mean and covariance estimations are here based on the full number of observations with complete data for each (pair of) variable(s). Imputation methods involve replacing missing values by estimates such as by the mean of observed values in the attribute, called mean imputation. Single imputation imputes a single value treating it as known, whereas multiple imputation replaces missing values by two or more values representing a distribution of possibilities. In multiple imputation, missing data are filled in an arbitrary number of times to generate different complete datasets to be analysed, and results are combined for inference. Finally, in Bayesian network learning, the Expectation Maximization (EM) algorithm [2] is often applied, which iteratively optimises parameters in order to find the maximum likelihood estimate, assuming the missing data is missing at random (MAR). The Structural EM algorithm (SEM) [3] combines this standard EM algorithm with structure search for model selection.

The variant of the structural EM algorithm that is used in this case study can be described as follows (see Algorithm 1 for an overview). Let $\mathbf{d}$ be a dataset over the set of random variables $\mathbf{V}$. Assume that $\mathbf{o}$ is part of the dataset that is actually observed, i.e., $\mathbf{o} \subseteq \mathbf{d}$. Furthermore, we denote the missing data by $\mathbf{h}$, i.e., $\mathbf{d} = \mathbf{o} \cup \mathbf{h}$, and $\mathbf{o} \cap \mathbf{h} = \varnothing$. The SEM algorithm aims to find a model from the space of Bayesian network models over $\mathbf{V}$, denoted by $\mathcal{M}$, such that each model $M \in \mathcal{M}$ is parametrised by a vector Θ^M defining a probability distribution $\mathbb{P}(\mathbf{V} : M, \Theta^M)$. To find a model in case of missing values, the complete data likelihood $\mathbb{P}(\mathbf{H}, \mathbf{O} \mid M)$ is estimated. The algorithm iteratively maximises the expected Bayesian network model score optimised by the score-based algo-

rithm. First the posterior parameter distributions, given the currently best model structure and observed data, are computed. In the expectation step, these distributions are used to compute the expected complete dataset, imputing missing values with their most probable values, also sometimes called *hard EM*. During the maximization, the currently best model structure is updated using a tabu structure learning algorithm, using the imputed data from the expectation step. Then parameter learning gives new distributions to be used as input for the next expectation step. To perform the first expectation, an initial network structure is given as input to the algorithm. In case a maximum number of iterations is reached or in case of convergence, the Bayesian network model is returned.

3 Description of the Data and Methodology

The experiments in this intervention case study aim to analyse performance of different methods to handle missing values and to learn the Bayesian network model for given intervention data in order to compare its results to previous analyses. This section describes the data, preprocessing phase, magnitude of the missing data problem and the approach to determine a suitable method in order to analyse the data by Bayesian network learning. The raw research data that has been collected during the Active Plus intervention was provided to the authors and is described in the first subsection.

3.1 Data acquisition and description

The raw research data has mostly been collected via questionnaires and consists of determinants, external factors, measurements of physical activity and intervention-related information at different time-slots, starting with a baseline measurement before the participant receives the intervention [14]. For example, the validated self-administrated Dutch Short Questionnaire to Assess Health Enhancing Physical Activity (SQUASH) is included in the questionnaires as subjective measurement of physical activity [15]. Figure 1 illustrates the intervention outline including moments of receiving intervention content and of measurement in time [12]. There is a distinction between control, intervention basic and intervention-plus groups, representing the intervention condition. This condition determines whether a participant receives an intervention or not and if environmental content is included in the intervention with additional information such as opportunities to be physically active in the own environment. Within these main groups, content is further personalised based on characteristics of participants, for example state of behaviour change (stage) measured at baseline or age. Since in the analyses in this article intervention content is proxied by a few main characteristics, this personalisation is beyond the focus of this article [12].

As depicted in Figure 1, data has been collected at 4 time-slots; at the baseline (before receiving the intervention, T0) and, to measure intervention effects, 3 (T1), 6 (T2) and 12 (T3) months after the baseline. About 1258 variables have

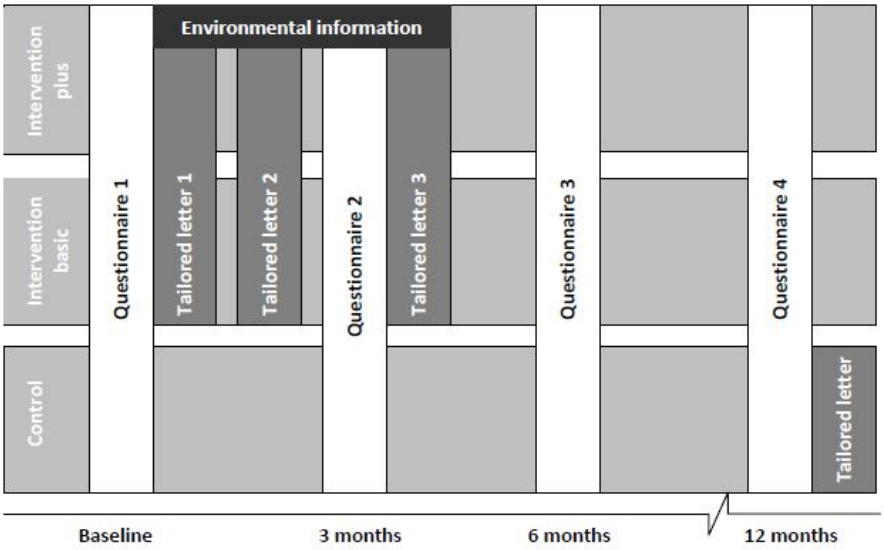


Fig. 1: Outline intervention program including moments of measurement [12].

been measured for a sub-population being a random sample of 1976 adults aged 50 and older. Measurements are at item-level of detail, where an item is a specific measurement, for example a question in the questionnaire. In preprocessing rules, it is described how concepts are calculated from item data in order to perform analyses at a higher level of abstraction.

3.2 Data preprocessing and concept design

The raw data is preprocessed, according to rules to integrate data from different studies and to aggregate, by calculating concepts from the raw data at item-level of detail, as mentioned in previous subsection. This subsection describes assumptions and decisions made during data preprocessing phase and rules to calculate the concepts included in analyses in this article.

In general, concepts are calculated by the mean or sum of items taking into account a maximum percentage of items allowed to be missing, except from a few concepts calculated using predefined formulas. In particular, the SQUASH-outcome measure, which is the number of minutes per week of moderate to intensive physical activity, is calculated in a standardised way [12]. In case more than 25 percent of the items are missing, the concept value is assumed to be missing. Besides these aggregation rules, preprocessing rules contain decisions about recalculation of raw data values to unipolar scale.

This article focuses on a selection of the data measured in the Active Plus intervention and, as already mentioned, analyses are performed at concept-level. The selection consists of data about the main determinants of physical activity

Concept	Number of items	T0	T1	T2	T3
Condition: intervention	1	X			
Condition: environment	1	X			
SQUASH outcome measure	-	X	X	X	X
Self-efficacy	10	X	X		
Attitude(-pros)	9	X	X		
Attitude(-cons)	7	X	X		
Intrinsic motivation	6	X	X		
Intention	3	X	X	X	X
Commitment	3	X	X	X	
Strategic planning	10	X	X	X	X
Action planning	6	X	X	X	
Coping planning	5	X	X	X	
Habit	12	X		X	X
Social modelling	1	X	X	X	
Social support	1	X	X		

Table 1: Overview of concept-level variables included in case study.

behaviour, including some social-related determinants, the main outcome measure from the SQUASH questionnaire and some variables indicating the intervention content the participant receives. As described, the intervention content that an individual participant has received is personalised and proxied in the analyses. The proxy of the intervention content is represented in the data by intervention condition variables, which thus play a central role in analyses. Table 1 gives an overview of these and all other concepts included in this articles analyses, indicating the number of item-level variables the concept variable aggregates and at which moments in time the concept is measured. Note that the number of items for the SQUASH outcome measure is not indicated since it is calculated by standard rules.

3.3 Missing data analysis

A significant part of this case study consists of the evaluation of ways to handle missing data values. This subsection illustrates the magnitude of the missing data problem in the case study and determines which methods are appropriate to be evaluated.

A total of 39 variables being concepts at certain moments in time are selected as subset for analyses. Table 2 demonstrates the number of missing values out of 1976 observations for each of the included concept-level variable. Since the time dimension is crucial to analyse intervention effects and, as can be seen in Table 2, more than a fourth of the values are missing for measurements after the baseline, applying pairwise deletion would result in an immense loss of information. Furthermore, the number of complete observations is for the selection of concepts 360 out of 1976 in total, meaning that applying list-wise deletion would neglect a large part of the dataset. Since deletion methods are not appropriate to deal with the missing data in this case study, we resort to the remaining methods for dealing with missing data, i.e., mean imputation and the SEM algorithm described in Section 2.3, are applied and results are compared.

Concept	Timeslot	Number of missing values (out of 1976)
Condition: intervention	T0	8
Condition: environment	T0	8
SQUASH outcome measure	T0	3
	T1	518
	T2	565
	T3	628
Self-efficacy	T0	229
	T1	638
Attitude(-pros)	T0	149
	T1	587
Attitude(-cons)	T0	167
	T1	597
Intrinsic motivation	T0	325
	T1	690
Intention	T0	141
	T1	571
	T2	654
	T3	748
Commitment	T0	31
	T1	531
	T2	573
Strategic planning	T0	156
	T1	601
	T2	652
	T3	661
Action planning	T0	182
	T1	604
	T2	686
Coping planning	T0	192
	T1	621
	T2	668
Habit	T0	136
	T2	633
	T3	662
Social modelling	T0	532
	T1	915
	T2	952
Social support	T0	68
	T1	561

Table 2: Overview of number of missing values in included concepts.

3.4 Approach

This subsection discusses how a suitable method for handling missing data is determined in order to model the intervention data. To perform experiments,

Handling missing data	Mean log-likelihood	95% Confidence Interval
Mean imputation	-4779	[-4832;-4726]
SEM algorithm	-4127	[-4183;-4071]

Table 3: Results of cross-validation analysis for missing data methods.

the bnlearn package in R is used for Bayesian network learning [10]. Source code has been made publicly available[5].

In the comparison of the methods to handle missing data values evaluated in this article, we apply discrete dynamic Bayesian networks for preprocessed data that is discretised by manually creating intervals meaningful in the health psychology field. The models are learnt by the tabu search algorithm optimising the BIC score (see Section 2.2). In the intervention study at hand only system missing values occur, for example, in case a participant has not answered a specific question in the questionnaire or if the maximum amount of items allowed to be missing is exceeded. The methods evaluated both apply imputation where missing values are substituted by (maximum likelihood) estimators during the structure learning phase, namely mean imputation and the structural EM algorithm, introduced in Section 3.3. These two methods are compared by means of comparing the mean test-set log-likelihood using k-fold cross-validation (with $k = 10$).

Finally, a linear Gaussian temporal Bayesian network model for the Active Plus intervention data is constructed from the preprocessed selection of data by learning the network structure using SEM. It was chosen to learn a continuous network rather than a discrete one to prevent possible loss of information from the discretisation process. In order to evaluate significance of edges, a bootstrap analysis is applied. Edges that are identified in most bootstrap samples and in the original network are considered stable findings in the following.

4 Results

This section describes the performance comparison of the methods applied to handle missing values. Furthermore, the learnt Bayesian network to model the Active Plus data is presented and results are compared to previous analyses of relations between determinants in the study by Van Stralen et al. [13].

4.1 Comparison Bayesian network missing data strategy

Table 3 demonstrates the mean log-likelihood over the folds resulting from applying the implemented cross-validation algorithm to the selected methods for handling missing data.

The cross-validation analysis shows that the structural EM algorithm significantly outperforms mean imputation to handle missing data, since the mean

[5] `https://github.com/SCMWTUM/Active4life-datascience.git`

Model	Statistics	
Optimal Bayesian network	#nodes	39
	# arcs	188
	# undirected arcs	0
	Average markov blanket size	19.90
	Average neighbourhood size	9.64
	Average branching factor	4.82
Averaged Bayesian network	#nodes	39
	# arcs	170
	# undirected arcs	0
	Average markov blanket size	17.54
	Average neighbourhood size	8.72
	Average branching factor	4.36

Table 4: Statistics Bayesian network model versus averaged counterpart.

log-likelihoods over the folds significantly differ at 5% confidence level. In the next subsection, the learnt model is presented and results are compared to those from previous analyses.

4.2 Comparison of Bayesian network model to previous analyses

Figure 2 shows the union of the temporal Bayesian network model learnt by the tabu search algorithm, applying SEM and optimising BIC score, and the result of bootstrapping (which we call *averaged model*). A comparison of these models shows that only 149 edges appear in both models represented by black edges in Figure 2, 21 only in the averaged model represented by red edges, and 39 only in the optimal model learnt from the data represented by blue dashed edges. Table 4 gives the summary statistics of the temporal Bayesian network model learnt and its averaged counterpart and indicates that model complexity is decreased in the averaged model. This suggests that most edges are stable, but not in all cases. Quite some edges appear to be unstable, which is something that should be analysed further in future.

Compared to previous analyses, the Bayesian network model provides a more complete insight in the complexity of mechanisms influencing physical activity behaviour. Previously, mediation analyses have shown that factors such as social modelling, self-efficacy and intention are significant mediators of the intervention influencing physical activity behaviour. In Figure 3, a fragment of the stable part of the averaged model (Figure 2) is shown that includes these previously proven significant determinants, intervention effects, and effects on physical activity. It also includes coefficients, which represent the maximum likelihood estimators of parameters of the Gaussian conditional density distribution of variables given their parents. This part of the network suggests that intervention effect on physical activity levels is mainly mediated by influencing habit and intention, and the extension in which environmental components are added to the intervention does

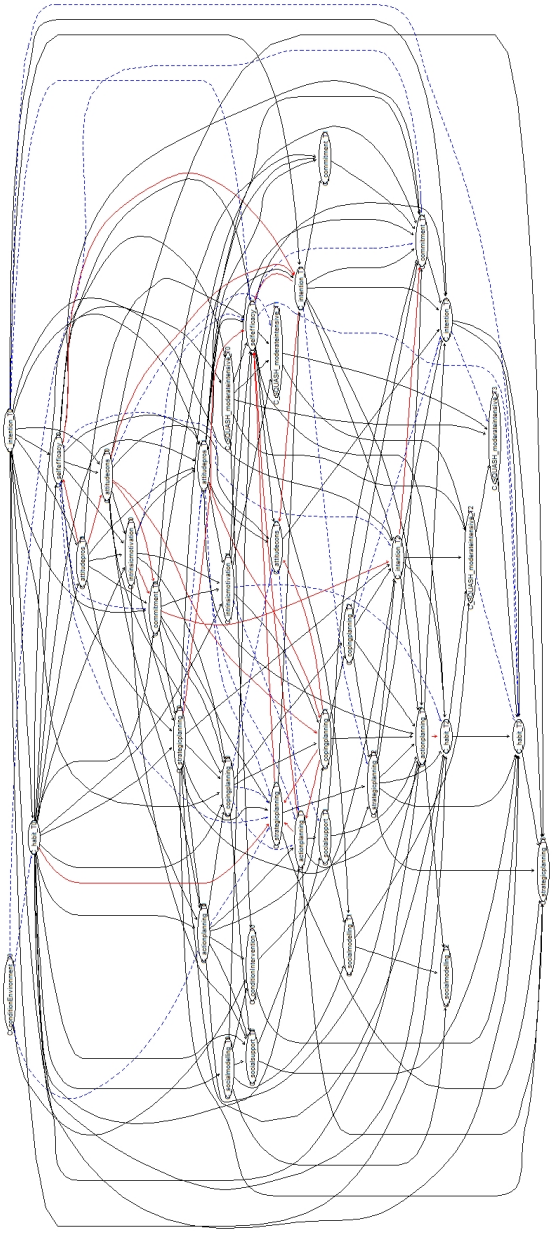


Fig. 2: Averaged model learnt from bootstrapping, which includes the false positives (in blue) and false negatives (in red) from the model learnt for the original dataset.

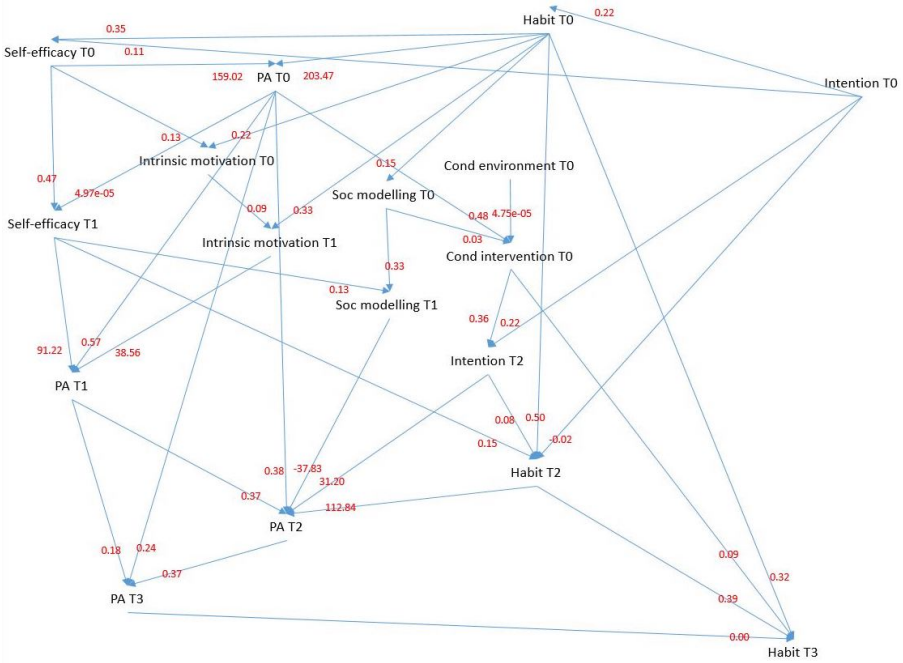


Fig. 3: Selected subgraph of the averaged model.

not significantly influence physical activity nor its determinants. Furthermore, there is a distinction between determinants of physical activity in the short (T1) and in the long (T2 and T3) run. In the short run, effects on physical activity are mainly determined by self-efficacy and intrinsic-motivation, which mediates effects of habit and self-efficacy. In the long-run, social modelling, intention and habit are important, where habit has the strongest correlation with long run physical activity levels.

Looking at intervention effect analysis, comparing these results to previous results by [13], significant influence on social modelling and self-efficacy in the short run is not demonstrated in the network. Looking at mediator effects on physical activity, [13] has not found that intrinsic motivation is relevant in the short run, whereas in the Bayesian network this determinant does have a significant direct influence on physical activity level in the short run. Also, where in previous analyses results show significant influence of the environmental extension on physical activity and determinant levels, this relationship is not found in the Bayesian network model. All in all, the added value of applying the Bayesian network model compared to traditional analyses is that the model provides new in-depth information relevant for understanding the working mechanisms of the intervention. For example, intrinsic motivation might mediate effects of other determinants found in [13], such as attitude-cons, on physical activity in the

short run, which might explain that the Bayesian network model leads to the difference compared to classic mediator analyses that has been found.

To conclude this section, differences found between previous results and results from the Bayesian network model are explored and possible explanations are provided. First of all, previously-found important mediators of intervention effects on physical activity such as social modelling, self-efficacy and intention are confirmed by the network model. In the network, habit is also a significant mediator. [13] did not include this determinant in analyses, so no comparison can be made with respect to habit being a significant mediator of intervention effects on physical activity. An important difference is that [13] found differences between effects in groups of participants having received environmental content and those who did not receive this extension. In the Bayesian network, those differences are not found. However, taking into account uncertain edges, there are some interesting correlations of the environmental extension with, for example, commitment at T2. Further analyses could explore these relations in order to explain the differences. There are also differences found with respect to intervention effects on determinants and mediation effects on physical activity. [13] has not found intrinsic motivation being a significant mediator, whereas the network model shows that the effect of self-efficacy on physical activity is both direct and mediated by intrinsic motivation. The network explores the mechanism in which self-efficacy influences physical activity, so that intrinsic motivation emerges as mediator. In the network, the intervention does not have a direct effect on social modelling nor on self-efficacy. This can be explained by looking at the whole model, where for example the intervention influences intention, which is correlated with action planning that is again correlated with social modelling. In this way, some determinants previously-found to be influenced by the intervention directly, are indicated in the network to be influenced via other determinants. Hence, the network provides a more in-depth view in the dependencies and the structure in which determinants and physical activity are influenced by the intervention.

5 Discussion and Conclusions

In this article, the Bayesian network modelling technique has been applied to an e-health intervention case study to potentially better understand relations between determinants of physical activity, since this technique has not been applied often in this field and traditional analyses are not sufficient to reveal the dependence structure between determinants. The magnitude of the major challenge of missing values in performing machine learning in real-world studies in general is examined for this case study and is shown to be of such an order that conventional methods to handle missing values cannot be used. The performance of different methods to handle missing data in Bayesian network modelling (i.e. mean imputation and the structural EM algorithm), considered to be appropriate in this case study, has been evaluated. Although the comparison between the mean imputation and structural EM method is not very novel from a machine

learning point of view, it has been carried out to evaluate their performances in this specific context. Also, since this modelling technique has not yet often been applied in this research field, its added value compared to more classic analyses in health psychology is evaluated by learning a Bayesian network for the case study and comparing its results to those of previous analyses on the same data.

Analysis of missing data in the case study dataset demonstrates that the magnitude of this problem causes methods to handle missing data based on deletion to be inappropriate, since this would result in a significant loss of information for this type of data. Two suitable methods, i.e., mean imputation and the structural EM algorithm, have been compared and we show that applying the structural EM algorithm leads to the best results in terms of fitting the data when learning a Bayesian network model for intervention data. The model learnt for the case study data applying this algorithm to handle missing values, suggests that the intervention does influence physical activity behaviour, that some concepts do not play a direct role influencing this behaviour or are not directly influenced significantly by the intervention and, most importantly, that there is some structure of how determinants explain this behaviour. Furthermore, there is some room for improvement with respect to increase confidence in some relationships in the model. Focusing on significant edges in a submodel, some differences regarding significant direct correlations are found compared to previous analyses. In brief, it can be concluded that applying Bayesian networks to e-health intervention study data provides more insight in the complexity of how interventions cause behavioural change (physical activity) and therewith are a useful technique to better understand dependence mechanisms of determinants of behaviour change.

In future work, analyses in this article could be extended for example by evaluating other imputation methods to be implemented in the structural EM algorithm, such as a distribution over values instead of imputing the value with highest probability (*soft EM*). From a technical perspective, we will also consider exploring constraint-based structure learning algorithms, other score-based algorithms, alternative parameter learning algorithms or alternative model selection criteria. From the application perspective, future research could further elaborate on the structure, in which determinants are related to each other and physical activity, and on the differences found in the Bayesian network model compared to previous (regression) analyses. Also, it would be interesting to perform analyses in more detail by using item variables in order to clarify the correlations between concepts found in the learnt network model presented in this paper. Finally, a combined model could be designed for an integrated dataset including measurements from several different e-health intervention studies, on different sub-populations, in order to examine if the general model yields different or additional results compared to the submodels for a smaller amount of data from single studies. However, even with data from a single study, this paper shows that exploring the differences between results from previous analyses and from the Bayesian network model, the network provides a more complete and in-depth insight in dependency structures. More specifically, the network reveals

relations between variables where a variable influences another via a third one. In previous analyses, only some of the hypothetical mediator effects are explored by regression analyses. Hence, our results provide new opportunities to analyse and confirm our findings using traditional statistical methods.

Acknowledgements This work is part of the research programme Active4Life with project number 546003005, which is financed by ZonMw.

References

1. Brug, J., van Assema, P., Lechner, L.: Gezondheidsvoorlichting en gedragsverandering Een planmatige aanpak. Koninklijke Van Gorcum, Assen, 9th edn. (2017)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)
3. Friedman, N.: The Bayesian structural EM algorithm. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. pp. 129–138 (1998)
4. Ji, Z., Xia, Q., Meng, G.: A review of parameter learning methods in Bayesian network. Advanced Intelligent Computing Theories and Applications **9227**, 9–12 (2015)
5. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
6. Murphy, K.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis, UC Berkeley (2002)
7. Nakai, M., Ke, W.: Review of the methods for handling missing data in longitudinal data analysis. International Journal of Mathematical Analysis **5**(1), 1–13 (2011)
8. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA (1988)
9. Schwarz, G., et al.: Estimating the dimension of a model. The annals of statistics **6**(2), 461–464 (1978)
10. Scutari, M.: Package 'bnlearn'. Bayesian network structure learning, parameter learning and inference, R package version 4.4 **1** (2019)
11. Scutari, M., Graafland, C.E., Gutiérrez, J.M.: Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms. International Journal of Approximate Reasoning **115**, 235–253 (December 2019)
12. van Stralen, M.M., Kok, G., de Vries, H., Mudde, A.N., Bolman, C., Lechner, L.: The active plus protocol: systematic development of two tailored physical activity interventions for older adults. BMC Public Health **8** (2008)
13. van Stralen, M.M., de Vries, H., Bolman, C., Mudde, A.N., Lechner, L.: Exploring the efficacy and moderators of two computer-tailored physical activity interventions for older adults: a randomized controlled trial. Annals of Behavioral Medicine **39**(2), 139–150 (2010)
14. van Stralen, M.M., de Vries, H., Mudde, A.N., Bolman, C., Lechner, L.: Determinants of initiation and maintenance of physical activity among older adults: a literature review. Health Psychology Review **3**, 147–207 (2009)
15. Wendel-Vos, G., Schuit, A., Saris, W., Kromhout, D.: Reproducibility and relative validity of the short questionnaire to assess health-enhancing physical activity. J Clin Epidemiol **56**, 1163–1169 (2003)

Dutch Humor Detection by Generating Negative Examples

Thomas Winters[0000−0001−7494−2453] and Pieter Delobelle[0000−0001−5911−5310]

Dept. of Computer Science; Leuven.AI
KU Leuven, Belgium
firstname.lastname@kuleuven.be

Abstract. Detecting if a text is humorous is a hard task to do computationally, as it usually requires linguistic and common sense insights. In machine learning, humor detection is usually modeled as a binary classification task, trained to predict if the given text is a joke or another type of text. Rather than using completely different non-humorous texts, we propose using text generation algorithms for imitating the original joke dataset to increase the difficulty for the learning algorithm. We constructed several different joke and non-joke datasets to test the humor detection abilities of different language technologies. In particular, we compare the humor detection capabilities of classic neural network approaches with the state-of-the-art Dutch language model RobBERT. In doing so, we create and compare the first Dutch humor detection systems. We found that while other models perform well when the non-jokes came from completely different domains, RobBERT was the only one that was able to distinguish jokes from generated negative examples. This performance illustrates the usefulness of using text generation to create negative datasets for humor recognition, and also shows that transformer models are a large step forward in humor detection.

Keywords: Computational Humor · Humor Detection · RobBERT · BERT model

1 Introduction

Humor is an intrinsically human trait. All human cultures have created some form of humorous artifacts for making others laugh [5]. Most humor theories also define humor in function of the reaction of the perceiving humans to humorous artifacts. For example, according to the popular incongruity-resolution theory, we laugh because our mind discovers that an initial mental image of a particular text is incorrect, and that this text has a second, latent interpretation that only becomes apparent when the punchline is heard [25,24]. To determine that something is a joke, the listener thus has to mentally represent the set-up, followed by detecting an incongruity caused by hearing the punchline, and resolve this by inhibiting the first, non-humorous interpretation and understanding the second interpretation [8,14]. Such humor definitions thus tie humor to the

abilities and limitations of the human mind: if the joke is too easy or too hard for our brain, one of the mental images might not get established, and lead to the joke not being perceived as humorous. As such, making computers truly and completely recognize and understand a joke would not only require the computer to understand and notice the two possible interpretations, but also that a human would perceive these as two distinct interpretations. Since the field of artificial intelligence is currently nowhere near such mental image processing capacity, truly computationally understanding arbitrary jokes seems far off.

While truly understanding jokes in a computational way is a challenging natural language processing task, there have been several studies that researched and developed humor detection systems [26,20,17,6,2,29,1]. Such systems usually model humor detection as a binary classification task where the system predicts if the given text is a joke or not. The non-jokes often come from completely different datasets, such as news and proverbs [20,36,6,2,7,1]. In this paper, we create the non-joke dataset by using text generation algorithms designed to mimic the original joke dataset by only using words that are used in the joke corpus [31]. This dataset thus substantially increases the difficulty of humor detection, especially for algorithms that use word-based features, given that coherence plays a more important role in distinguishing the two. We use the recent RobBERT model to test if its linguistic abilities allow it to also tackle the difficult challenge of humor detection, especially on our new type of dataset. As far as the authors are aware, this paper also introduces the first Dutch humor detection systems.

2 Background

2.1 Neural Language Models

Neural networks perform incredibly well when dealing with a fixed number of features. When dealing with sequences of varying lengths, recurrent connections to previous states can be added to the network, as done in recurrent neural networks (RNN). Long short-term memory (LSTM) networks are a variety of RNN that add several gates for accessing and forgetting previously seen information. This way, a sequence can be represented by a fixed-length feature vector by using the last hidden states of multiple LSTM cells [15]. Alternatively, if a maximum sequence length is known, the input size of a neural network could be set to this maximum, and e.g. allow for using a convolutional neural network (CNN).

Entering text into a recurrent neural network is usually done by processing the text as a sequence of words or tokens, each represented by a single vector from pre-trained embeddings containing semantic information [21]. These vectors are obtained from large corpora in the target language, where the context of a token is predicted e.g. using Bag-of-Words (BOW) [21].

BERT The BERT model is a powerful language model that improved many state-of-the-art performances on NLP tasks [11]. It is built using a transformer encoder stack consisting of self-attention heads to create a bidirectional language

model [28]. These attention mechanisms allow BERT to distinguish different meanings for particular words based on the context by using contextualized embeddings. For example, even though the word *"stick"* could be both a noun as well as a verb, normal word embeddings assign the same vector to both meanings.

BERT is trained in a self-supervised way by predicting missing words in sentences, and predicting if two randomly chosen sentences are subsequent or not. After this pre-training phase, additional heads can be fine-tuned on particular datasets to classify full sentences, or to classify every token of a sentence. The model exhibits large quantities of linguistic knowledge (e.g. for resolving coreferences, POS tagging, sentiment analysis) and achieved state-of-the-art performance on many different language tasks. This model later got critically re-evaluated and improved in the RoBERTa model, which uses a revised training regime [19].

RobBERT RobBERT [10] is a recent Dutch language model trained using the RoBERTa regime [19] on the Dutch OSCAR corpus section [22]. Like RoBERTa, RobBERT also outperforms other Dutch language models in a wide range of complex NLP tasks e.g. coreference resolution and sentiment analysis [10].

2.2 Humor Detection

Humor is not an easy feat for computational models. True humor understanding would need large quantities of linguistic knowledge and common sense about the world to know that an initial interpretation is being revealed to be incompatible with the second, hidden interpretation fitting the whole joke rather than only the premise. Many humor detection systems use hand-crafted (often word-based) features to distinguish jokes from non-jokes [26,20,17,2]. Such word-based features perform well when the non-joke dataset is using completely different words than the joke dataset. From humor theory, we know that the order of words matter, since stating the punchline before the setup would only cause the second interpretation of the joke to be discovered, making the joke lose its humorous aspect [24]. Since word-based humor detectors often fail to capture such temporal differences, more contextual-aware language models are required to capture the true differences between jokes and non-jokes.

Using a large pre-trained model like the recent BERT-like models is thus an interesting fit for the humor detection task. One possible downside is that these models are not well suited for grasping complex wordplay, as their tokens are unaware of relative morphological similarities, due to the models being unaware of the letters of the tokens [3]. Nevertheless, BERT-like models have performed well on English humor recognition datasets [29,1].

Recently, several parallel English satirical headline corpora have been released for detecting humor, which might help capture subtle textual differences that create or remove humor [30,16]. Lower resource languages however usually do not have access to such annotated parallel corpora for niche tasks like humor detection. While there has been some Dutch computational humor research [31,32,33], there has not been any published research about Dutch humor detection, nor are there any public Dutch humor detection data sets available.

2.3 Text Generation for Imitation

There are many types of text generation algorithms. The most popular type of text generation algorithms use statistical models to iteratively predict the next word given previous words, e.g. n-gram based Markov models or GPT-2 and GPT-3 [23,4]. These algorithms usually generate locally coherent text [31]. A common downside is that the output tends to have globally different structures than the training data (e.g. much longer or shorter), or even break (possibly latent) templates of the dataset [34].

Dynamic Templates Templates can be seen as texts with holes, which are later filled, and thus enforce a global textual structure. One approach for learning these is the dynamic template algorithm (DT), which is designed to replace context words with other, grammatically similar words [31]. It achieves this by analyzing the part-of-speech (POS) tags in the dynamic template text and replaces these words with context words with the same POS tags. It prioritizes low unigram-frequency words, as these are usually key words determining the context of the text. This way, the dynamic template algorithm generates a large variety of more nonsensical versions of given texts, using only words from the corpus.

3 Data

3.1 Collecting Datasets

We collected a Dutch joke dataset by combining the jokes found on Kidsweek[1], DeBesteMoppen[2] and LachJeKrom[3]. This resulted in a dataset of **3235 jokes**.

For the non-joke datasets, we first collected several datasets inspired by the type of datasets used in English humor detection, namely proverbs and news [20,36,6,2,7,29,1]. The proverbs dataset originates from the Dutch proverbs Wikipedia page and contains **1887 proverbs**. The news dataset are **3235 headlines** uniformly sampled from the 100K Dutch news headlines dataset [37].

3.2 Negative Generation: Generating Non-Jokes from Jokes

Since news and proverbs use completely different words and structures, there is a need for a new type of challenging dataset for humor recognition that uses non-jokes that are close to jokes. Given the fragile nature of a joke, changing several important words usually turn the joke into a non-humorous text. We propose a new type of dataset for humor detection by generating negative examples by automatically imitating the joke dataset. The dynamic template algorithm is a right fit for this, as it will not change the global structure like Markov models might do and is less prone to plagiarising large parts of the training corpus [31].

[1] https://www.kidsweek.nl/moppen
[2] https://www.debestemoppen.nl/
[3] https://www.lachjekrom.com/

The DT algorithm creates absurd, but globally similar texts, by grammatically similar words into another joke. For example, the joke *"Wat is groen en plakt aan de muur? Kermit de sticker!"*[4] was turned into the non-joke *"Wat is groen en telefoneert aan de muur? Kermit de spin!"*[5].

We chose the same parametrisation used in the original paper (see Appendix A) [31]. The resulting non-jokes thus only use words from the jokes dataset, with comparable frequencies, and still have similar grammatical structures, albeit nonsensical content. This way, language classifiers that just learn which words are more common in jokes (e.g. *"oen"*, *"Jantje"*, *"blond"*...) will be at a disadvantage compared to models that have better insight in the semantic coherence of a joke. Another advantage of this method for parallel corpus creation is that it is easily extensible to other lower resource languages.

4 Evaluation

We devised two types of learning tasks for detecting humor in these new datasets. The first is the classic humor detection task with binary labels representing joke and non-joke. The second is a pairwise humor detection task, where given a joke and a non-joke, the algorithm needs to detect which of the two is a joke.

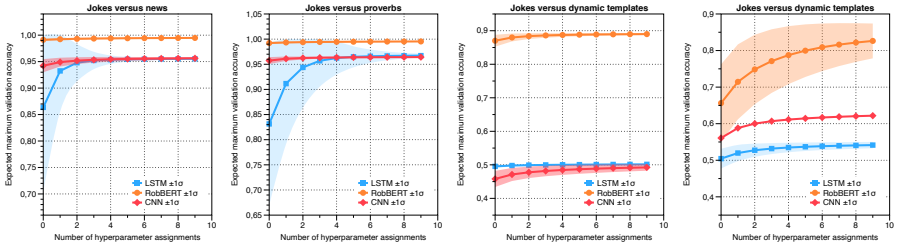
4.1 Models

We compare four different models[6], namely a Naive Bayes classifier with the TF-IDF of 3000 (1,3)-grams as features, an LSTM with Dutch word embeddings [27], a CNN with two convolutional layers and max pooling on Dutch word embeddings [18], and RobBERT [10]. The use of LSTMs and CNNs allows us to compare the RobBERT model with the previous generation of neural language models.

4.2 Classification Experiment

In this binary classification experiment, the models classifies a given text as a joke or a non-joke. We compared three different datasets, comparing jokes with news, with proverbs, and with generated jokes using the dynamic template algorithm. We performed a random hyperparameter search with 10 runs for the LSTM, CNN, and RobBERT. The full search space and other hyperparameters are listed in the Appendix in Table 2 and Table 3. In addition, we use these *random* hyperparameter trials to estimate the maximum validation accuracy [12]. This allows us to compare performance without it being caused by a computational budget favoring one model. Figure 1 shows these estimates for the validation accuracy for all three datasets. For both the news (Figure 1a) and proverbs (Figure 1b), both the

[4] *"What's green and adheres to the wall? Kermit the Sticker"*, pun on *"kikker"* (*"frog"*)
[5] *"What's green and telephones on the wall? Kermit the Spider"*
[6] The code, models, data collectors and demo are available on https://github.com/twinters/dutch-humor-detection.



(a) News (b) Proverbs (c) Single DT (d) Pairwise DT

Fig. 1: Estimated maximum validation accuracy [12] in function of the number of hyperparameter trials for the LSTM, CNN, and RobBERT models.

CNN and the LSTM-based models perform a couple of percentage points below the RobBERT model. More notably, the RobBERT model consistently achieves a validation accuracy around 99%, whilst the LSTM has a higher variance than both the CNN model and RobBERT. This indicates that the LSTM-based model is less robust to suboptimal hyperparameter assignment.

From these randomized trials, we select the best-performing model using the validation accuracy and evaluate on the held-out test set. The results are presented in Table 1. The baseline, Naive Bayes, performs quite poorly, with the results being no better than random on all three datasets. This is surprising, given that this method has been used successfully for humor detection in English for similar types of datasets, albeit often using handcrafted features instead of words [20,17,2]. This shows that a classifier using only token features is insufficient for all three Dutch humor datasets. The LSTM and CNN models recognise about 94% for the simple datasets However, they both fail at distinguishing between jokes and non-jokes generated with dynamic templates. This indicates that despite using Dutch word embeddings, these models likely still relies on vocabulary differences or the small lengths news and proverbs tends to have.

Finetuning RobBERT gives us a testing accuracy of 98.8% and 99.6% on *news* and *proverbs*, respectively, and 89.2% on the more challenging task with dynamic templates. This shows that our newly created dataset is indeed more challenging than using non-jokes from completely different domains. Interestingly, RobBERT's false positives contain many jokes with only limited replaced words, or that still made semantically coherent sense, e.g. the joke *"Hoe heet de broer van Bill Mars? Bill Twix!"*[7], only had one replacement (*"Bruno" to "Bill"*), retaining some of the joke's humor. RobBERT's higher performance than the other neural networks also illustrates the advantage of pre-trained language models for detecting semantic coherence in jokes, or at least distinguishing it from semantically incoherent non-jokes generated by the DT algorithm, either of which are useful properties. To get more grasp on this, we classified all elements of the news and the proverbs dataset using the finetuned RobBERT model for the

[7] *"What's the name of Bill Mars' brother? Bill Twix!"*

Table 1: Classification results for the held-out test set on three datasets versus the Jokes dataset. For the accuracy, we additionally report the 95% CI.

					Dynamic template			
	News		**Proverbs**		**Single**		**Pairwise**	
Model	ACC [%]	F_1 [%]	ACC [%]	F_1 [%]	ACC [%]	F_1 [%]	ACC [%]	F_1 [%]
Naive Bayes	51.0 ± 3.1	49.3	60.2 ± 3.5	50.5	49.9 ± 3.1	49.9	-	-
LSTM	94.0 ± 1.5	94.0	94.4 ± 1.6	94.1	46.8 ± 3.1	35.2	47.9 ± 4.4	32.4
CNN	93.6 ± 1.5	93.6	94.1 ± 1.7	93.6	47.4 ± 3.1	46.3	58.6 ± 4.4	58.5
RobBERT	$\mathbf{98.8} \pm 0.7$	**98.8**	$\mathbf{99.6} \pm 0.4$	**99.6**	$\mathbf{89.2} \pm 1.9$	**89.1**	$\mathbf{82.51} \pm 3.4$	**82.5**

jokes versus dynamic template single setting. This input data was thus completely out-of-domain for this model. We found that 93.23% of the news and 73% of the proverbs were labeled as a joke by this model, indicating that at least for such relatively short strings, the DT model might rely on topical or semantic coherence to recognise humor.

4.3 Pairwise Classification Experiment

We additionally perform an experiment where the joke and their non-joke counterpart, generated by the DT algorithm, are directly compared in a pairwise fashion. The model thus has to recognise which one is more humorous than the other, opening the way for humor preference learning algorithms [13].

We evaluated an LSTM model with two separate recurrent layers with trainable Dutch word embeddings that are concatenated before a fully connected layer, which was also used for argument classification [9]. We evaluated a CNN model following a similar approach, with the same base architecture as in the previous experiment. For RobBERT, we are using the same setup and hyperparameters, and feed the model both texts simultaneously, separated by the separator token.

In Table 1, we can see that LSTMs are still unable to distinguish jokes from generated non-jokes, and CNNs only seeing a small performance boost in the pairwise case over the single case, illustrating the advantage of using such a challenging dataset. RobBERT on the other hand is performing reasonably well but surprisingly loses some accuracy compared to the single classification case. This is likely due to relatively more of the jokes being truncated to fit its input size limit, given that two texts are now fitted into the same input space.

5 Future Work

One way to improve the humor detection performance could be finding better ways of generating joke-like non-jokes, thus further increasing the difficulty of the dataset. The DT algorithm is prone to occasional grammatical errors, which the models might pick up and use to just recognize grammatical errors, rather than recognize jokes.

These new humor detection algorithms also pave way for new humor generators, e.g. using a generate-and-test approach [35]. RobBERT could even fulfill two roles in such a generator, e.g. using a genetic algorithm that uses a pairwise joke detection head as tournament selection, and the word masking head to mutate the genomes. Such a generator could also be useful in a collaborative setting where the humor comparator suggests better ways of phrasing a joke by subtly changing it e.g. by rearranging a potential punchline word to occur later.

6 Conclusion

We created three datasets for humor detection specifically for Dutch and proposed a new way to make more challenging humor detection datasets. We hypothesized that currently popular approaches, like discerning news or proverbs, can rely on recognizing domain-specific vocabularies instead of the semantic coherence that makes jokes funny. We illustrated this by constructing several models for humor detection on these new datasets; where we found that previous technologies indeed are not or barely able to distinguish jokes from similar non-jokes. For a more modern architecture like RobBERT, the performance is only slightly lower for the generated non-jokes. This shows that the generated negatives dataset is indeed more challenging, and that transformer models are a step in the right direction for humor detection given their context-sensitivity. These datasets and findings open the way for interesting new, more context-aware Dutch joke detection and generation algorithms.

Acknowledgements

Thomas Winters is a fellow of the Research Foundation-Flanders (FWO-Vlaanderen). Pieter Delobelle was supported by the Research Foundation - Flanders under EOS No. 30992574 and received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Annamoradnejad, I.: Colbert: Using bert sentence embedding for humor detection. arXiv preprint arXiv:2004.12765 (2020)
2. van den Beukel, S., Aroyo, L.: Homonym detection for humor recognition in short text. In: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. pp. 286–291 (2018)
3. Branwen, G.: Gpt-3 creative fiction (2020)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
5. Caron, J.E.: From ethology to aesthetics: Evolution as a theoretical paradigm for research on laughter, humor, and other comic phenomena. Humor **15**(3), 245–281 (2002)

6. Cattle, A., Ma, X.: Recognizing humour using word associations and humour anchor extraction. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1849–1858 (2018)
7. Chen, P.Y., Soo, V.W.: Humor recognition using deep learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 113–117 (2018)
8. Deckers, L., Buttram, R.T.: Humor as a response to incongruities within or between schemata. Humor: International Journal of Humor Research (1990)
9. Delobelle, P., Cunha, M., Massip Cano, E., Peperkamp, J., Berendt, B.: Computational ad hominem detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 203–209. Association for Computational Linguistics, Florence, Italy (Jul 2019)
10. Delobelle, P., Winters, T., Berendt, B.: RobBERT: a Dutch RoBERTa-based language model. arXiv preprint arXiv:2001.06286 (2020)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
12. Dodge, J., Gururangan, S., Card, D., Schwartz, R., Smith, N.A.: Show your work: Improved reporting of experimental results. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2185–2194. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
13. Fürnkranz, J., Hüllermeier, E.: Pairwise preference learning and ranking. In: European conference on machine learning. pp. 145–156. Springer (2003)
14. Gibson, J.: A good sense of humor is a sign of psychological health. quartz (2016)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
16. Hossain, N., Krumm, J., Gamon, M.: " president vows to cut¡ taxes¿ hair": Dataset and analysis of creative text editing for humorous headlines. arXiv preprint arXiv:1906.00274 (2019)
17. Kiddon, C., Brun, Y.: That's what she said: double entendre identification. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. pp. 89–94 (2011)
18. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (Oct 2014)
19. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs] (Jul 2019)
20. Mihalcea, R., Strapparava, C.: Making computers laugh: Investigations in automatic humor recognition. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 531–538. Association for Computational Linguistics (2005)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

22. Ortiz Suárez, P.J., Sagot, B., Romary, L.: Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In: 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7). Cardiff, United Kingdom (Jul 2019)
23. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8) (2019)
24. Ritchie, G.: Developing the incongruity-resolution theory. Informatics Report Series (10 1999)
25. Suls, J.M.: A two-stage model for the appreciation of jokes and cartoons: An information-processing analysis. The psychology of humor: Theoretical perspectives and empirical issues **1**, 81–100 (1972)
26. Taylor, J.M., Mazlack, L.J.: Computationally recognizing wordplay in jokes. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 26 (2004)
27. Tulkens, S., Emmery, C., Daelemans, W.: Evaluating unsupervised dutch word embeddings as a linguistic resource. In: Chair), N.C.C., Choukri, K., Declerck, T., Grobelnik, M., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (may 2016)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)
29. Weller, O., Seppi, K.: Humor detection: A transformer gets the last laugh. arXiv preprint arXiv:1909.00252 (2019)
30. West, R., Horvitz, E.: Reverse-engineering satire, or "paper on computational humor accepted despite making serious advances". In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 7265–7272 (2019)
31. Winters, T.: Generating philosophical statements using interpolated markov models and dynamic templates. In: 31st European Summer School in Logic, Language and Information Student Session Proceedings. pp. 181–189. Riga, Latvia, ESSLLI (Aug 2019)
32. Winters, T.: Generating dutch punning riddles about current affairs. 29th Meeting of Computational Linguistics in the Netherlands (CLIN 2019): Book of Abstracts (Jan 2019)
33. Winters, T.: Modelling mutually interactive fictional character conversational agents. In: Proceedings of the 31st Benelux Conference on Artificial Intelligence (BNAIC 2019) and the 28th Belgian Dutch Conference on Machine Learning (Benelearn 2019). vol. 2491. CEUR-WS (2019)
34. Winters, T., De Raedt, L.: Discovering textual structures: Generative grammar induction using template trees. Proceedings of the 11th International Conference on Computational Creativity pp. 177–180 (2020)
35. Winters, T., Nys, V., De Schreye, D.: Towards a general framework for humor generation from rated examples. Proceedings of the 10th International Conference on Computational Creativity pp. 274–281 (2019)
36. Yang, D., Lavie, A., Dyer, C., Hovy, E.: Humor recognition and humor anchor extraction. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2367–2376 (2015)
37. Yeh, C.L., Loni, B., Hendriks, M., Reinhardt, H., Schuth, A.: Dpgmedia2019: A dutch news dataset for partisanship detection. arXiv preprint arXiv:1908.02322 (2019)

A Dynamic Template parametrisation

The parameters used for the dynamic template algorithm to generate the non-jokes are a maximum word frequency of the 62% percentile, and minimum number of replacement of at least one replacement for every 25 characters, and three randomly sampled jokes for context words

B Hyperparameter space

Table 2: The hyperparameter space used for the LSTM and CNN models with Dutch word embeddings.

Hyperparameter	**LSTM model**	**CNN model**
adam_epsilon	10^{-8}	10^{-8}
fp16	False	False
hidden_dimension	$i \in \{8, 16, 32, 64, 128\}$	—
learning_rate	$[10^{-3}, 10^{-1}]$	$[10^{-3}, 10^{-1}]$
pooling	—	max
convolutional_layers	—	2
kernel_size.	—	3
max_grad_norm	1.0	1.0
num_train_epochs	15	15
batch_size	64	64
seed	1	1
dropout	0.1	0.1

Table 3: The hyperparameter space used for finetuning RobBERT.

Hyperparameter	**Value**
adam_epsilon	10^{-8}
fp16	False
gradient_accumulation_steps	$i \in \{1, 2, 3, 4\}$
learning_rate	$[10^{-6}, 10^{-4}]$
max_grad_norm	1.0
max_steps	-1
num_train_epochs	3
per_device_eval_batch_size	8
per_device_train_batch_size	8
seed	1
warmup_steps	0
weight_decay	$[0, 0.1]$

Transaction Cost Allocation in Industrial Symbiosis: A Multiagent Systems Approach

Vahid Yazdanpanah[1], Devrim M. Yazan[2], and W. Henk M. Zijm[2]

[1] Agents, Interaction and Complexity Group
University of Southampton, Southampton, UK
V.Yazdanpanah@soton.ac.uk
[2] Department of Industrial Engineering and Business Information Systems
University of Twente, Enschede, The Netherlands
{D.M.Yazan,W.H.M.Zijm}@utwente.nl

Abstract. This paper discusses the dynamics of Transaction Cost (TC) in Industrial Symbiosis Institutions (ISI) and provides a fair and stable mechanism for TC allocation among the involved firms in a given ISI. In principle, industrial symbiosis, as an implementation of the circular economy paradigm in the context of industrial relation, is a practice aiming at reducing the material/energy footprint of the firms. The well-engineered form of this practice is proved to decrease the transaction costs at a collective level. This can be achieved using information systems for identifying potential synergies, evaluating mutually beneficial ones, implementing the contracts, and governing the behavior of the established relations. Then the question is: "*how to distribute the costs for maintaining such an information system in a fair and stable manner?*" We see such a cost as a *collective transaction cost* and employ an integrated method rooted in cooperative game theory and multiagent systems research to develop a fair and stable allocation mechanism for it. The contribution is twofold: in developing computational multiagent methods for capturing the dynamics of transaction costs in industrial symbiosis and in presenting a novel game-theoretic mechanism for its allocation in industrial symbiosis institutions.

Keywords: Multiagent Systems · Applied AI · Practical Applications of Multiagent Techniques · Agent-Based Computational Economics · Industrial Symbiosis.

1 Introduction

Industrial symbiosis is a transitional business model to shift from a linear economy paradigm towards implementing the concept of circular economy in the context of industrial relations/networks. In principle, the aim is to facilitate the circulation of reusable resources among the network members [8,18,32]. Realizing such a form of collaboration requires methods for identifying potential matches [17], evaluating them to generate mutually beneficial instances [35], implementing the cost-sharing schemes in bilateral contracts [33], and decentralized governance of the established relations [34]. As elaborated in [36] problematic situations occur when we move from bilateral relations to multilateral forms in multiagent industrial symbiosis. Dealing with such problems requires: (1) developing practical methods able to capture collective-level concepts, such as collectively realized transaction costs, (2) practice-oriented semantics

to reason about the link between individual (firm-level) concerns and the collective (institution-level) attributes, and (3) implementable mechanisms to guarantee desirable collective attributes with no harm to firm-level concerns. This work aims to address this gap by focusing on the nature of collective transaction costs in industrial symbiosis and developing a contextualized allocation mechanism that guarantees game-theoretical fairness and stability properties.

As originally introduced by [31], Transaction Costs (TC) play a key role in the establishment and stability of different forms of contractual relations. This is also the case for industrial symbiosis relations, both in bilateral and multilateral forms. For instance, when an industrial cluster manager aims to share the collective costs for maintaining the shared environments among the cluster members, she should take into account the firm-level concerns. One general approach is to see the contribution of each firm to the collective as a measure for making such cost/benefit allocation decisions [3]. In the case of IS, while we have such contribution-aware tools for allocating costs in bilateral IS relations [33], they fail to guarantee some basic properties such as collective rationality (related to whether firms have any incentive to leave the collaboration) on a multilateral network level [36]. In practice, such a disadvantage results in inefficient deployment of an IS management platform in real-life industrial symbiosis networks.

To develop an efficient technique for dealing with this problem, it is crucial to understand the context and see what are transaction costs in IS. In multiagent industrial symbiosis, the main elements that contribute to the transaction cost are costs involved in market/partner searching, negotiation costs, and the relation enforcement cost [7]. We argue that in modern industrial symbiosis, and thanks to (online) IS information systems/platforms, the total IS transaction cost boils down to costs for establishment and maintenance of the information system which is responsible for handling the searching/matching process, for supporting/automating the negotiations, and for synthesizing the required enforcement measures. Then the question is about finding methods to distribute this total cost among the involved firms, such that it would be fair with respect to each firm's contribution. That means that, in a given IS networks, after a basic payment for getting involved in the platform, firms that gain more (as a result of exploiting the potentials in the platform) are expected to pay more for upcoming transaction costs. In other words: "*with more power comes more responsibility*". This perspective supports the so called *fairness* notions—proposed in cooperative game theory—that agents' individual benefit/cost share oughts to reflect their contribution to the collective benefit/cost [13,37]. While the idea to take into account each agent's contribution provides a basis for allocating the costs in multiagent industrial symbiosis, we lack methods for defining the value of each and every coalition of firms, involved in the network[1]. Such an input is crucial for applying standard fair allocation mechanisms (e.g., the notion of the Shapley value [25,22]). In response, developing a method that characterizes IS as a game is the first objective of this work. The result of this first step will be a game-form that in turn enables the application of game theoretic solution concepts for allocating the transaction cost. Technical details will follow in Section 2.

[1] In the game-theoretic language, the characteristic function of the cooperative multiagent industrial symbiosis is not well-defined with respect to the context of industrial symbiosis and its constraints.

To formulate a realistic game, based on which the transaction cost can be allocated, one question that is crucial to address is "*what is the nature of transaction cost in IS practices?*". In the IS literature [16,35], costs for operationalizing IS are categorized as costs for transporting the resource, for treating them by means of recycling/preparation processes, and finally transaction costs as discussed above. In principle, transportation and treatment costs are based on the realization of physical facts and interactions[2]. For instance, the physical distance between the firms is one of the key measures that determines the transportation cost and hence can be a fair basis for allocating the collective transportation cost among the involved firms (e.g., see [30]). For treatment costs—as the total cost that results from recycling, drying, and categorizing—we also have physical processes that consume resources (e.g., electricity and diesel cost of a recycling plant). However, the transaction cost is not based on physical facts, but mainly is the aggregation of costs of institutional acts, e.g., negotiation costs and corresponding costs for establishing monitoring/enforcement mechanisms. These acts are on the one hand required to enable or keep track of physical processes (hence have a *physical* connotation) and on the other hand are related to the structure of interfirm connections (hence have an *institutional* nature). Therefore, it would be reasonable to consider a form of dual-nature *physical-institutional IS value* as a basis for allocating the collective transaction cost among the involved firms. This way of capturing the institutional dimension of the transaction cost is in-line with the original studies on this notion, as presented in [31,28]. Although the line of reasoning seems straightforward, capturing such a perspective in a practical tool for allocating the collective transaction costs in multiagent industrial symbiosis is an open problem. (As highlighted by [36], some standard allocation mechanisms are inapplicable for IS implementation due to operational complexities.) Next, we elaborate on our approach on tailoring formal multiagent methods for modeling industrial symbiosis as an institution and for developing an operationally feasible transaction cost allocation mechanism.

We see multiagent industrial symbiosis as a practical manifestation of (well-designed) industrial institutions and aim to model it using game-theoretic methods, able to capture both the physical and the institutional contributions of involved firms[3]. Such a formal approach enables employing institutional economics methods for guaranteeing

[2] To reason about the essence of transaction costs, we use the terminology of [24] and form our perspective based on the categorization of *physical* and *institutional* facts and their corresponding acts. Physical facts are about the valuation of a variable in the observable world (e.g., the *quantity* of a resource or the *distance* between two firms) while institutional facts are about the invisible world of concepts, definable in a given context (e.g., the fact that a resource *is needed* or that a firm *is powerful*). Physical acts may change the value of physical facts while institutional acts affect the institutional facts. We abstract from the reasoning structures that relate the two disjoint sets of facts/acts. See [34] for further elaborations on such a conceptualization in the context of IS.

[3] We use the term *institution* as a general reference to a collective of entities behaving in a systematic manner, under an emerged or established coordination mechanism. Then a well-designed institution is one in which the mechanism is engineered such that some (collectively desirable) properties hold [23]. This would be distinguishable from the stronger notion of *organization* where an explicit representation of *roles*, organizational *structures*, and interaction *protocols* [1,5] is needed.

desirable properties at the collective level, e.g., see [36] for how a regulatory agent can influence the feasibility of multiagent industrial symbiosis by means of incentive engineering techniques.

In relation to the focus of this work, i.e., the collective transaction cost and its allocation among the involved firms, the *fairness* property is concerned with capturing the contributions of firms as a basis for cost allocation. While physical acts/facts (e.g., distance and quantifiable used energy) determine a base for computing fair allocation of (physical) transportation/treatment costs, a fair allocation of the collective transaction cost calls for a notion to capture the institutional contribution of agents as well. To that end, we provide a formal account of the transaction cost economics in industrial symbiosis (based on computational organization theory and industrial institutions). This results in the introduction of the notion of *industrial symbiosis index* as a measure for capturing the physical and institutional contribution of firms. In turn, this notion will be a base for developing a fair and stable transaction cost allocation mechanism. Finally, we elaborate on potential questions to be solved using the provided methodological foundation.

2 Multiagent Industrial Symbiosis Institutions

To model IS institutions, we build upon a graphical representation of cooperative games—also known as *graphical* games [19] or *graph-restricted* games [21]. Such a representation is a natural choice as it reflects the established relations among the firms and allows the application of standard fair division methods for sharing the collective transaction cost, among the members of the institution[4]. As a first step, we use graph-theoretic notions to determine a realistic characteristic function for the game-theoretic representation of industrial symbiosis. Then, adding an allocation mechanism results in our formal notion of industrial symbiosis institution.

We recall basic game theoretic notions and the definition of graphical games based on [20,6,21].

Cooperative Games: A (transferable utility) cooperative game on a finite set of agents Γ is a tuple $\langle \Gamma, f \rangle$ where the game's characteristic function $f : 2^{\Gamma} \mapsto \mathbb{R}$ is such that $f(\emptyset) = 0$.

Graphical Games: A graphical (transferable utility) cooperative game on a finite set of agents/vertices Γ is a triple $\langle \Gamma, W, f \rangle$ where W is a $|\Gamma| \times |\Gamma|$ real-valued weight matrix (representing the weights of edges between vertices in Γ) and the game's W-restricted characteristic function $f^W : 2^{\Gamma} \mapsto \mathbb{R}$ is such that $f(\emptyset) = 0$. We say f is restricted to W as it determines the value of any coalition $S \subseteq \Gamma \setminus \emptyset$ with respect to W. Such a general formalization allows further tailoring in the context of industrial symbiosis.

Allocation Mechanisms: For a given cooperative game $\mathcal{G} = \langle \Gamma, f \rangle$, a (single-point) allocation mechanism $\mathcal{M}$ maps a real-valued tuple $\mathcal{M}(\mathcal{G}) \in \mathbb{R}^{|\Gamma|}$ to the pointed game. The i-th element of the allocation tuple $\mathcal{M}(\mathcal{G}) = \langle a_1, a_2, \ldots, a_{|\Gamma|} \rangle$ is the *share* of agent

[4] Through the course of this work, we may refer to firms as *agents*. This is to see any industrial symbiosis institution as an environment that supports the collaborative interaction of a set of autonomous decision-makers in charge of the involved firms.

$i \in \Gamma$ according to $\mathcal{M}$ and with respect to $\mathcal{G}$. The term *share* can be interpreted—with respect to the context—as the amount to be *paid* or *gained* by i. We later discuss various properties that such a mechanism can hold or bring about.

To determine how the transaction cost can be allocated among the firms based on their physical and institutional contributions, we take the graph that represents the established symbiotic relations and obtained cost reductions as an input. (Note that the reasoning about such a cost allocation takes place in a retrospective manner and after the establishment of IS relations.) Given such a graph, we formulate a game-theoretical representation that in turn results in inducing the physical as well as the institutional contribution of individual firms.

Definition 1 (IS Graph). *An IS graph is a tuple $\langle \Gamma, W \rangle$, where Γ is the set of vertices, representing $|\Gamma|$ firms and W is the $|\Gamma| \times |\Gamma|$ matrix of positive real valued weights, representing the cost reduction values. There exists a weighted undirected edge between distinct firms $i, j \in \Gamma$, representing their established symbiotic relation, only if $W_{i,j} \neq 0$. Moreover, for any $i \in \Gamma$ we have that $\sum_{j \in \Gamma} W_{i,j} > 0$ (connected) and that $W_{i,i} = 0$ (loop-freeness).*

To have a concise and contextualized representation, we don't require an explicit set of edges as it could be derived based on W. The same holds for requiring the graph to be loop-free and connected. Basically, loops and disconnected firms can be excluded as in such cases the transaction cost (hence its allocation) is meaningless. This results in a realistic representation in which unfeasible relations/edges (which otherwise could be represented by negative or zero weights) are excluded. In the context of IS, $W_{i,j}$ reflects the realized net benefit—in terms of collectively obtained cost reductions—of the symbiotic relation between firms i and j on a given (quantity of) resource r. As discussed in [35], such a collective benefit can be computed by deducing the total operational cost of the relation (for treatment and transportation of r) from the total traditional costs (for discharging r on the provider side of the relation and purchasing traditionally-used inputs—substituted by r in the realized relation—on the receiver side). The W graph would be the basis for formulating both the physical IS game (reflecting obtainable benefits) and the institutional game (modeling the institutional power of firms in the cluster).

Definition 2 (Physical IS Game). *A graphical physical IS game is a triple $\langle \Gamma, W, v \rangle$, where $G = \langle \Gamma, W \rangle$ is an IS graph and for any group of firms $S \subseteq \Gamma$ with $|S| > 1$, the characteristic function $v(S)$ is equal to $\frac{1}{2} \sum_{i,j \in S} W_{i,j}$. By convention, for any S with $|S| \leq 1$, $v(S) = 0$. Then in the normalized characteristic function, denoted by $\bar{v}$, we have that $\bar{v}(S) = v(S)/v(\Gamma)$.*

Example 1. To demonstrate the applicability of our approach, we use a case study (adopted from a realistic industrial cluster[5]). See Figure 1 for an illustration of the IS graph. In this graph, the value on each edge reflects the benefit (in terms of cost

[5] The adopted case is one of the successful implementations of IS networks, investigated in a European project. Due to confidentiality concerns, we omitted the company names and modified some values.

reductions) that resulted from the symbiotic relation, realized between the nodes that it connects. While such values represent the physical dimension of an IS practice, the structure of the graph is what we later use to formulate the institutional importance of each node/firm.

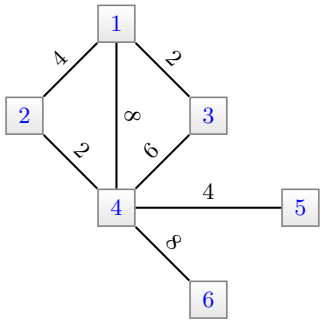


Fig. 1. Connectivity graph of the involved firms in the IS cluster. Each node represents a firm and the value on each edge represents the amount of cost reduction, obtained as a result of the collaboration between the firms that the edge connects. Values are presented as *utils*. A util can be interpreted as any form of transferable utility, e.g., a util may be equal to 100$.

The value of any singleton or empty coalition is 0 while for any S with $|S| > 1$, we calculate the value by simply adding up the weights of the edges that connect any two member of S. For instance, $v(\{1,2,3,4\}) = 22$ and $v(\Gamma = \{1,\ldots,6\}) = 34$. For the same coalitions, the normalized values are respectively $\frac{22}{34}$ and $\frac{34}{34}$. These normalized values will be later employed for aggregation of the physical game with a game that represents the institutional power of firms in such clusters.

Below, we present game theoretic properties of the physical IS games.

Proposition 1 (Properties). *Let $\mathcal{G} = \langle \Gamma, W, v\rangle$ be a physical IS game. Then: (1) for any coalitions $S \subset T$, we have that $v(S) \leq v(T)$ (monotonicity); (2) for any disjoint coalitions S and T, we have that $v(S \cup T) \geq v(S) + v(T)$ (super-additivity); (3) for any coalitions S and T, we have that $v(S \cup T) + v(S \cap T) \geq v(S) + v(T)$ (convexity/super-modularity).*

Proof. (1) imagine a firm i in $T \setminus S$. If i is connected to a member of S or $T \setminus S$, it contributes to the value of T. Otherwise, it has the added value of 0. In both cases, part (1) is true. *(2)* if there exist a direct edge connecting a member of S to a member of the disjoint coalition T, then $v(S \cup T)$ increases; otherwise, it is equal to $v(S) + v(T)$, thanks to non-negative weights. *(3)* In case the two sets are disjoint, it follows from part (2). Otherwise, the two coalitions have nonempty intersection with either positive or zero value. □

These properties result in the following practical result that the collective value of the grand coalition Γ can be shared among the firms such that no coalition has an incentive to defect the collaboration.

Theorem 1 (Nonempty Core). *Let $\mathcal{G} = \langle \Gamma, W, v \rangle$ be a physical IS game. Then there exists an allocation mechanism $\mathcal{M}$ such that in $\mathcal{M}(\mathcal{G}) = \langle a_1, a_2, \ldots, a_{|\Gamma|} \rangle$, we have (1) $\sum_{i \in \Gamma} a_i = v(\Gamma)$ (Efficiency) and (2) for any $S \subseteq \Gamma$, $\sum_{i \in S} a_i \geq v(S)$ (Coalitional Rationality).*

Proof. The two clauses in this theorem axiomatize the notion of core-nonemptiness [20]. Then, the convexity property (in Proposition 1) in combination with the well-established Bondareva-Shapley theorem [4,26] ensures the nonemptiness of the core, and accordingly the existence of a mechanism to generate the allocation. □

In addition to the physical game—based on which we can induce the *physical* contributions that firms can bring about through a bilateral exchange of resources—the next step is to present a basis for capturing the *institutional* contribution of firms. For such a purpose, we cannot rely on cost reduction values (obtained thanks to operationalizing the relations) basically because transaction costs are non-operational, but have an institutional nature. To that end, we employ the notion of *closeness centrality* adopted from the literature on communication networks [2] to capture the institutional power of firms, and as the basis for defining the characteristic function of the IS *institutional* game. In a graph on the set of vertices Γ, the closeness centrality of a vertex $i \in \Gamma$, denoted by $\mathfrak{C}(i)$, is equal to $\frac{|\Gamma|-1}{\sum_{j \in \Gamma \setminus \{i\}} d(i,j)}$ where the distance function $d : \Gamma \times \Gamma \mapsto \mathbb{N}^+$ returns the shortest distance between i and j. More explicitly, $d(i,j) = d(j,i)$ returns the minimum number of edges passed to reach j from i. Recalling the presented IS graph in Figure 1, we have that $\mathfrak{C}(1) = \frac{5}{7}$, $\mathfrak{C}(2) = \mathfrak{C}(3) = \frac{5}{8}$, $\mathfrak{C}(4) = \frac{5}{5}$, and $\mathfrak{C}(5) = \mathfrak{C}(6) = \frac{5}{9}$.

Then based on this notions, we formulate the institutional IS game as follows.

Definition 3 (Institutional IS Game). *A graphical institutional IS game is a triple $\langle \Gamma, W, \iota \rangle$, where $G = \langle \Gamma, W \rangle$ is an IS graph and for any group of firms $S \subseteq \Gamma$ with $|S| > 0$, the characteristic function $\iota(S)$ is equal to $\sum_{i \in S} \mathfrak{C}(i)$. By convention, $\iota(\emptyset) = 0$. Then in the normalized characteristic function, denoted by $\bar{\iota}$, we have that $\bar{\iota}(S) = \iota(S)/\iota(\Gamma)$.*

Due to the additive formulation of ι, we have the following properties for institutional IS games.

Proposition 2 (Properties). *Let $\mathcal{G} = \langle \Gamma, W, \iota \rangle$ be an institutional IS game. Then $\mathcal{G}$ is (1) monotonic and (2) convex/super-modular (defined analogously to Proposition 1).*

Proof. Given that $\mathfrak{C}(i)$ is non-negative for all $i \in \Gamma$ and the formulation of $\iota(S)$ as the summation of $\mathfrak{C}(i)$ for all $i \in S$, monotonicity is trivial. For convexity, it suffices to decompose $\iota(S \cup T)$. We have that $\iota(S \cup T)$ is equal to $\sum_{i \in S \cup T} \mathfrak{C}(i) = \sum_{j \in S} \mathfrak{C}(j) + \sum_{k \in T} \mathfrak{C}(k) - \sum_{l \in S \cap T} \mathfrak{C}(l)$, hence $\iota(S \cup T) + \iota(S \cap T) = \iota(S) + \iota(T)$, which satisfies the convexity condition. □

Immediate to this, we have the existence of an efficient and coalitionally rational allocation mechanism for any institutional IS game (parallel to Theorem 1 for physical IS games). Having both the physical and the institutional aspects of IS formalized in the game-theoretic language, we present the aggregated IS game as the summation of the normalized form of the two games.

Definition 4 (IS Game). *Let $G = \langle \Gamma, W \rangle$ be an IS graph, $\mathcal{G}_P = \langle \Gamma, W, v \rangle$ a physical IS game on G, and $\mathcal{G}_I = \langle \Gamma, W, \iota \rangle$ an institutional IS game on G. Then the graphical IS game is a triple $\langle \Gamma, W, \sigma \rangle$, such that for any group of firms $S \subseteq \Gamma$, the characteristic function $\sigma(S)$ is equal to $\bar{v}(S) + \bar{\iota}(S)$.*

Then, as a corollary to Propositions 1 and 2 and Theorem 1, we immediately deduce that any IS game preserves the properties presented in Proposition 2, hence has a non-empty core (analogous to Theorem 1).

Corollary 1. *For any IS game $\langle \Gamma, W, \sigma \rangle$, the set of efficient and coalitionally rational allocation mechanisms is non-empty.*

In other words: the normalized versions of both games satisfy the presented properties and linear aggregation preserves them. In some application domains, one may opt for various forms of linear aggregations, i.e., to employ $\sigma = \alpha\bar{v} + \beta\bar{\iota}$ (for integer-valued positive α and β). We later highlight that due to the linearity of the allocation mechanism that we employ, our results remain valid in such a generalization of the problem.

Following the presented perspective in [35], an industrial symbiosis institution consists of a group of firms, a structure that specifies the outcome of collaboration among potential coalitions (in the group), and mechanism(s) responsible for coordinating the institution. Such mechanisms are basically in charge to guarantee some desirable properties in the institution. In our case—in industrial symbiosis institutions—the aim could be to ensure the *stability* of the institution (i.e., that no firm or group of firms has an incentive to defect the collaboratively profitable institution), the *fair allocation* of the collectively obtained benefits (such that the contribution of firms is reasonably reflected in their individual shares), or ideally to bring about both *fairness and stability*. In a general form, an industrial symbiosis institution is defined as:

Definition 5 (Industrial Symbiosis Institution). *Let Γ be a set of firms, $\mathcal{G}$ an IS game among firms in Γ, and $\mathfrak{M}$ a set of value allocation mechanisms. Then an industrial symbiosis institution is defined as triple $\mathcal{I} = \langle \Gamma, \mathcal{G}, \mathfrak{M} \rangle$.*

In brief, this is to see an IS institution as an IS game under mechanisms in charge of distributing the collective values. (Note that we do not fix the allocation mechanism but take a set $\mathfrak{M}$.) This would be to distribute collectively obtainable benefits as well as collective operational costs for establishment and maintenance of the institution. The latter category corresponds to the focus of this work for allocating transaction costs in industrial symbiosis. In the next section, we present an allocation mechanism—corresponding to the notion of Shapley value and the Myerson value in graph-restricted games [25,21]—that satisfies both *fairness* and *stability* properties. We also elaborate on its computational complexity and tractability results.

3 A Fair Transaction Cost Allocation Mechanism

The main idea behind the *fair* allocation of values is to take into account the contribution of each agent to the collaborative group [20]. In industrial symbiosis management

platforms, the *collective* transaction cost reduces to costs for establishment and maintenance of the framework—as a dynamic e-market environment[6]. This calls for dynamic cost allocation methods, able to grasp the physical as well as institutional nature of each agent's contribution. Roughly speaking, following an initial payment for a basic membership (to get involved in this e-market framework) it is expected that a "*fair*" allocation of costs for further improvements, takes into account the contribution of each participant—in terms of their role/function in the formed industrial network. Having the game-theoretic formulation of an IS game (as an aggregation of the physical and institutional IS games), any proportion of the transaction cost that oughts to be shared among the firms—e.g., the total cost for updating the IT platform—can be distributed based on each firm's contribution to the IS game. A standard notion in computational economics, capturing the contribution of each agent in a cooperative setting, is the *Shapley value* [22,25]. Shapley's allocation method uniquely satisfies the so called *fairness* properties, which has high relevance for our domain of application in industrial organizations. In IS games, as a combination of normalized physical and institutional games among the firms, the Shapley value of a firm determines the extent of its power and influence in the institution. This value would be defined as what we call the firm's *IS index*.

Definition 6 (IS Index). *Let $\mathcal{G} = \langle \Gamma, W, \sigma \rangle$ be an IS game. Then for any arbitrary $i \in \Gamma$, the IS index, denoted by $\Phi_i(\sigma)$, is equal to $\sum_{S \subseteq \Gamma \setminus \{i\}} \frac{|S|!(|\Gamma|-|S|-1)!}{|\Gamma|!}(\sigma(S \cup \{i\}) - \sigma(S))$.*

Due to the characteristics of σ (as a reflection of both the physical as well as institutional aspects of IS) the introduced notion of *IS index* is a measure that reflects the power of a firm based on its connectivity to other firms in the network and also its operational contributions by bringing about cost reductions. (We later show that due to the graphical representation of the problem, the IS index can be formulated in a non-factorial manner which leads to a low computational complexity result.) This index forms a basis for allocating transaction costs such that a higher contribution determines a higher share (i.e., *"with more power comes more responsibility"*). This approach relies on the standard rationale in cooperative cost-sharing games that agents with higher potentials ought to pay the larger share of the costs in the collaborative practice [20]. Accordingly, given an external cost function $\tau(\Gamma)$ (equal to $\tau(S)$ for all $S \subseteq \Gamma$ with $|S| \geq 2$), determining the (to be shared) transaction cost[7] among the members of Γ, we present the following transaction cost allocation mechanism for IS institutions.

[6] There may exist other forms of *individual* transaction costs in such relations. For instance, some investments to calibrate the production process to enable accepting a waste-based resource. However, as the equipment remains property of the firm, it is unreasonable to consider such a cost as a collective (to be shared) transaction cost.

[7] Such a cost may consist of initial platform development costs, ongoing IT infrastructure maintenance, or extra personnel recruitment costs for updating the platform. τ is defined in a functional way merely to allow further extensions on dynamic formulations of the transaction cost function, e.g., as a temporal function of some required resources for maintaining an IS information system. In other words, $\tau(S)$ is undefined for $|S| \leq 1$ and is equal to a given value $\tau(\Gamma)$ otherwise. Our main question is on how to distribute this collectively defined value.

Definition 7 (TC Allocation for IS). *Let $\mathcal{G} = \langle \Gamma, W, \sigma \rangle$ be an IS game, $\Phi_i(\sigma)$ the corresponding IS index for any arbitrary $i \in \Gamma$, and $\tau(\Gamma)$ a given transaction cost value for Γ. We define the cost share of agent $i \in \Gamma$ as $T_i(\sigma, \tau) := \frac{\Phi_i(\sigma) \cdot \tau(\Gamma)}{\sigma(\Gamma)}$. Allocation $T(\sigma, \tau) = \langle a_1, \ldots, a_{|\Gamma|} \rangle$ with $a_i = T_i(\sigma, \tau)$ denotes the TC allocation for IS game $\mathcal{G}$ with respect to τ.*

This allocation, tailored and contextualized for the specific class of structured, graph-restricted industrial symbiosis games, (1) captures both the physical as well as the institutional aspects of this practice, (2) satisfies desirable fairness and stability properties (to be discussed next), and (3) is computationally tractable thanks to the graphical representation of the games (to be illustrated in Section 4). Having an industrial institution, *stability* and *fairness* are two properties insurable by means of well-designed mechanisms. In the case of the transaction cost allocation mechanism, stability is about (1) sharing the exact amount of the cost and (2) sharing such that no firm can benefit by defecting from the institution. On the other hand, fairness is a more complex property, concerned with (1) the symmetric contribution of firms to the institution, (2) the share of firms whose involvement are non-contributory, (3) the possibility to aggregate various institutions, and finally (4) the sharing of the exact total cost. Below, we provide a formal account of these properties in axiomatic forms—based on [20]—and investigate whether they are valid in case of our suggested cost allocation mechanism.

Proposition 3 (Fairness Axioms). *Let $\mathcal{I} = \langle \Gamma, \mathcal{G}, \mathfrak{M} \rangle$ be an industrial symbiosis institution where Γ is the set of firms and $\mathcal{G} = \langle \Gamma, W, \sigma \rangle$ is the IS game. For any transaction cost $\tau(\Gamma) > 0$ we have that $\mathfrak{M} = \{T(\sigma, \tau)\}$ (Definition 7) guarantees the following fairness axioms: (1) The collective transaction cost is efficiently allocated among the firms, formally, $\sum_{i \in \Gamma} T_i(\sigma, \tau) = \tau(\Gamma)$ (Efficiency); (2) The identities of the firms do not affect their share of the total transaction cost, formally, for $i, j \in \Gamma$, $T_i(\sigma, \tau) = T_j(\sigma, \tau)$ if for all $S \subseteq \Gamma \setminus \{i, j\}$ we have that $\sigma(S \cup \{i\}) = \sigma(S \cup \{j\})$ (Symmetry); (3) Any firm of which its contribution to any coalition is equal to its individual value, pays a transaction cost share proportional to its individual value, formally, for $i \in \Gamma$, $T_i(\sigma, \tau) = \frac{\sigma(\{i\}) \cdot \tau(\Gamma)}{\sigma(\Gamma)}$ if for all $S \subseteq \Gamma \setminus \{i\}$ we have that $\sigma(S \cup \{i\}) = \sigma(S) + \sigma(\{i\})$(Dummy Player); (4) For two IS games, an agent's transaction cost share in the aggregated game is equal to the summation of its share in each, formally, given an industrial game $\mathcal{G}' = \langle \Gamma, W', \sigma' \rangle$ and a corresponding transaction cost $\tau'(\Gamma) > 0$, we have that $T_i(\sigma + \sigma', \tau + \tau') = T_i(\sigma, \tau) + T_i(\sigma', \tau')$ (Additivity).*

Proof. Our notion of IS index measures the Shapley value of each firm i. Following the linearity of this Shapley-based value, we have that the allocation mechanism preserves all the four properties that the Shapley value uniquely possesses [20]. □

In general, fairness and stability are orthogonal—an allocation might be fair but not stable or the other way around. Below, we present an axiomatic account of stability and show their validity for the presented transaction cost allocation method.

Proposition 4 (Stability Axioms). *Let $\mathcal{I} = \langle \Gamma, \mathcal{G}, \mathfrak{M} \rangle$ be an industrial symbiosis institution where Γ is the set of firms and $\mathcal{G} = \langle \Gamma, W, \sigma \rangle$ is the IS game. For any transaction cost $\tau(\Gamma) > 0$ we have that $\mathfrak{M} = \{T(\sigma, \tau)\}$ (Definition 7) guarantees the following*

stability axioms: (1) The collective transaction cost is efficiently allocated among the firms, formally, $\sum_{i\in\Gamma} T_i(\sigma,\tau) = \tau(\Gamma)$ *(Efficiency); (2) No subgroup faces an economic incentive to deviate from the grand coalition and benefit by paying a lower share of the transaction cost, formally, for any coalition* $S \subseteq \Gamma$ *with* $|S| \geq 2$*, we have that* $\sum_{i\in S} T_i(\sigma,\tau) \leq \tau(S)$ *(Coalitional Rationality).*

Proof. The first part is valid (using the previous proposition). For the second part, the current formulation of the collective transaction cost requires that $\tau(S) = \tau(\Gamma)$ which if combined with the first clause, immediately satisfies the claim. □

In the generalized form, where the transaction cost function is defined for all potential coalitions, the convexity of τ would be required for coalitional rationality. Note that in case such a function was available in the first place, the mere problem on "*how to distribute the collective cost*" would evaporate, as its solution requires a single call to that function.

Thanks to the adoption of a Shapley-based index—and its linearity property—the fairness and stability properties will be preserved in the presented aggregated form of IS institutions and any general linear aggregation forms in which the importance of the physical and institutional contributions are weighted.

Proposition 5 (Generalizability). *Let* $\mathcal{I} = \langle \Gamma, \mathcal{G}, \mathfrak{M}\rangle$ *be an industrial symbiosis institution where* $\sigma = \alpha\bar{v} + \beta\bar{\iota}$ *is the characteristic function of* $\mathcal{G}$ *in terms of* v *and* ι*, the corresponding characteristic functions in the physical and institutional IS games, respectively. We have that* $\mathfrak{M} = \{T(\sigma,\tau)\}$ *(Definition 7) guarantees fairness and stability in* $\mathcal{I}$*.*

4 Reductions Towards a Tractable Algorithm

Although the presented Shapley-based IS index has desirable properties, its standard formulation leads to computationally expensive algorithms. Below, we present reductions that result in an alternative formulation for computing the IS index.

Lemma 1. *In a graphical physical IS game* $\mathcal{G}_P = \langle \Gamma, W, v\rangle$*, for any* $i \in \Gamma$ *we have that* $\sum_{S\subseteq\Gamma\setminus\{i\}} \frac{|S|!(|\Gamma|-|S|-1)!}{|\Gamma|!}(v(S\cup\{i\}) - v(S)) = \sum_{j\in\Gamma\setminus\{i\}} \frac{W_{i,j}}{2}$.

Proof. Based on the formulation of v, the value of any singleton coalition S is zero and for any coalition T with more than two members, the value is computed based on the summation of values that bilateral relations (established within T) bring about. In other words, the average marginal contribution of any firm to any T with more than two members is zero. The only set of coalitions to which a firm may have a contribution are two-member coalitions for which we have the results of [33] that the middle point of the core corresponds to the average marginal contribution. In our graph-restricted games, this value is equal to the Myerson value [21] and is equal to half of the summation of the values on the edges that are directly connected to i, i.e., $\sum_{j\in\Gamma\setminus\{i\}} \frac{W_{i,j}}{2}$. □

Next we present a reduction for computing the contributions in the institutional game.

Lemma 2. *In a graphical institutional IS game $\mathcal{G}_I = \langle \Gamma, W, \iota \rangle$, for any $i \in \Gamma$ we have that* $\sum_{S \subseteq \Gamma \setminus \{i\}} \frac{|S|!(|\Gamma|-|S|-1)!}{|\Gamma|!}(\iota(S \cup \{i\}) - \iota(S)) = \mathfrak{C}(i)$.

Proof. In the institutional game, i's contribution to any coalition is equal to its degree of closeness centrality. Then the dummy player property implies the claim. □

Based on these reductions, the transaction cost allocation is computationally tractable.

Theorem 2. *Let $\mathcal{I} = \langle \Gamma, \mathcal{G}, \mathfrak{M} \rangle$ be an industrial symbiosis institution where Γ is the set of firms and $\mathcal{G} = \langle \Gamma, W, \sigma \rangle$ is the IS game. For any transaction cost $\tau(\Gamma) > 0$, employing $\mathfrak{M} = \{T(\sigma, \tau)\}$ to compute the allocation $T(\sigma, \tau) = \langle a_1, \ldots, a_{|\Gamma|} \rangle$ is polynomial in time and space.*

Proof. We present a constructive proof by providing an algorithm (see Algorithm 1) that generates the allocation, of which we verify its correctness and subsequently prove the complexity claims.

Algorithm 1: TC Cost Allocation in IS

Input : IS Graph $G = \langle \Gamma, W \rangle$ with Γ as the indexed set of firms and W as the $|\Gamma| \times |\Gamma|$ weight matrix, Transaction Cost $\tau(\Gamma)$.

```
% Initialization
n ← |Γ|
Sum(G) ← ½ Σ_{i,j∈Γ} W_{i,j}
Cent(G) ← Σ_{i∈Γ} 𝔠(i)
T ← [T_1, ..., T_n] % n-Member Allocation Array
% Allocation
for i ∈ Γ do
    Sum(i) ← 0
    for j ∈ Γ \ {i} do
        Sum(i) ← Sum(i) + W_{i,j}/2
    end
    % Compute IS Index Φ_i(σ)
    Φ_i(v̄) ← Sum(i)/Sum(G)
    Φ_i(ῑ) ← 𝔠(i)/Cent(G)
    Φ_i(σ) ← Φ_i(v̄) + Φ_i(ῑ)
    % Compute Individual Transaction Cost T_i(σ, τ)
    T_i(σ, τ) ← Φ_i(σ)·τ(Γ)/2
    T[i] ← T_i(σ, τ)
end
return T
```

Correctness: In Algorithm 1, for each firm $i \in \Gamma$, the IS index $\Phi_i(\sigma)$ is equal to the Shapley value of i in the aggregated game (of the normalized physical and institutional games). Thanks to the additivity property, this would be equal to the aggregation of Shapley values in each game. Then, we rely on Lemma 1 and 2 for calculating the two

values. Finally, for computing individual transaction costs, we have that $\sigma(\Gamma) = 2$ as it is equal to $\frac{v(\Gamma)}{v(\Gamma)} + \frac{\iota(\Gamma)}{\iota(\Gamma)}$. *Space:* The required matrix of weights (representing the set of obtained cost reductions) is in $O(n^2)$ where n is the size of Γ. *Time:* For computing the IS indices, we have $O(n)$ on the big loop. Then in the physical game component, $Sum(i)$ is in $O(n)$ (a pass on the i-th row in W) and $Sum(G)$ is in $O(n^2)$ (a pass through the whole W). For the institutional part, computing $\mathfrak{C}(i)$ is reducible to finding the shortest paths [14] which is well-known to be in $O(n^3)$ [15]. □

To show the applicability of the developed method for allocating collective transaction costs among a cluster of firms Γ, we use the presented case in Example 1 and assume a total value $\tau(\Gamma)$ as the collective transaction cost, realized for an updating round in Γ's industrial symbiosis information system. Assuming $\tau(\Gamma) = 100$ simply results in percentage calculation for individual shares.

Following the steps in Algorithm 1, we have that $Sum(G) = 34$ and $Cent(G) = \frac{1027}{252}$. Then for each firm $i \in \{1, \ldots, 6\}$, to compute $\Phi_i(\bar{v})$ (as the physical component of $\Phi_i(\sigma)$), we calculate the summation of the weights on all the edges connected to i and divide it by $Sum(G)$. Thus we have: $\Phi_1(\bar{v}) = \frac{7}{34}$, $\Phi_2(\bar{v}) = \frac{3}{34}$, $\Phi_3(\bar{v}) = \frac{4}{34}$, $\Phi_4(\bar{v}) = \frac{14}{34}$, $\Phi_5(\bar{v}) = \frac{2}{34}$, $\Phi_6(\bar{v}) = \frac{4}{34}$. For each firm i, adding $\frac{\mathfrak{C}(i)}{Cent(G)}$ to $\Phi_i(\bar{v})$ results in its IS index $\Phi_i(\sigma) = \frac{13309}{34918}, \frac{4218}{17459}, \frac{9463}{34918}, \frac{11473}{17459}, \frac{3407}{17459}, \frac{4434}{17459}$ (respectively for firms 1 to 6). Finally, the transaction cost allocation T could be generated based on $\Phi_i(\sigma)$. We have that: $T_1(\sigma, \tau) = 19.06$, $T_2(\sigma, \tau) = 12.08$, $T_3(\sigma, \tau) = 13.55$, $T_4(\sigma, \tau) = 32.86$, $T_5(\sigma, \tau) = 9.76$, and $T_6(\sigma, \tau) = 12.70$.

Note that as we employ generic graph-/game-theoretical solution concepts as a basis for the developed algorithm, our results are neither sensitive to the distribution of the cost reduction values nor to the structure of the connectivity graph.

5 Concluding Remarks

Amid the institutional nature of transaction costs, to our knowledge, this work is the first proposal that translates Searle's well-established philosophy on institutional theory for the context of IS, takes it into practice for fair transaction cost allocation, and introduces a tractable algorithm for such a purpose. As a managerial decision support tool, the presented algorithm can be integrated into smart IS contracting and management frameworks to enable the automation of cost allocation procedures. For instance, as a suggested business model for IS clusters, firms would be expected to pay an initial membership fee and then be charged for further collective transaction costs based on the presented method—reflecting their operational as well as institutional contributions. As presented, this work has immediate applicability to support IS management by means of providing a fair and stable TC allocation mechanism. In addition, it opens new research directions. While we focused on generic IS, an interesting line of research is to investigate sub-classes of IS with respect to their graphical structures. For instance, in most bio-based IS practices, bio-refineries are in the center of the cluster due to their crucial role as a resource treatment facility. This results in tree-like structures or clusters of star graphs such that no two firms can implement an IS relation in the absence of a refinery. This calls for methods tailored to capture such contextual properties. To

this end, a combination of tree-like graphical games and dependence graphs [10] would be a suggested formal foundation. Another line for future work is to develop governance frameworks for IS. This is to enable monitoring of the organizational behavior and enforcing normatively desirable behaviors. For such a purpose, we aim to rely on the literature on norm-aware coordination [29,12,9] and address problems related to *organizational characterization* [11] of multiagent industrial symbiosis.

Acknowledgements. *SHAREBOX* [27], the project leading to this work, has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 680843.

References

1. Baldoni, M., Baroglio, C., Marengo, E.: Behavior-oriented commitment-based protocols. In: ECAI. vol. 215, pp. 137–142 (2010)
2. Bavelas, A.: Communication patterns in task-oriented groups. The Journal of the Acoustical Society of America **22**(6), 725–730 (1950)
3. Billera, L.J., Heath, D.C.: Allocation of shared costs: A set of axioms yielding a unique procedure. Mathematics of Operations Research **7**(1), 32–39 (1982)
4. Bondareva, O.N.: Some applications of linear programming methods to the theory of cooperative games. Problemy kibernetiki **10**, 119–139 (1963)
5. Van den Broek, E.L., Jonker, C.M., Sharpanskykh, A., Treur, J., et al.: Formal modeling and analysis of organizations. In: Proceedings of AAMAS-2005. pp. 18–34. Springer (2005)
6. Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational aspects of cooperative game theory. Synthesis Lectures on AI and Machine Learning **5**(6), 1–168 (2011)
7. Chertow, M., Ehrenfeld, J.: Organizing self-organizing systems: Toward a theory of industrial symbiosis. Journal of Industrial Ecology **16**(1), 13–27 (2012)
8. Chertow, M.R.: "Uncovering" industrial symbiosis. Journal of Industrial Ecology **11**(1), 11–30 (2007)
9. Conte, R., Dignum, F.: From social monitoring to normative influence. J. Artif. Soc. Soc. Simul. **4**(2) (2001)
10. Conte, R., Sichman, J.S.: Dependence graphs: Dependence within and between groups. Computational & Mathematical Organization Theory **8**(2), 87–112 (2002)
11. Coutinho, L.R., Brandão, A., Sichman, J.S., Hübner, J.F., Boissier, O.: A model-based architecture for organizational interoperability in open multiagent systems. In: Coordination, Organizations, Institutions and Norms in Agent Systems. pp. 102–113. Springer (2009)
12. Dastani, M., van der Torre, L., Yorke-Smith, N.: Commitments and interaction norms in organisations. Autonomous Agents and Multi-Agent Systems **31**(2), 207–249 (2017)
13. Deng, X., Papadimitriou, C.H.: On the complexity of cooperative solution concepts. Mathematics of Operations Research **19**(2), 257–266 (1994)
14. Eppstein, D., Wang, J.: Fast approximation of centrality. In: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms. pp. 228–229 (2001)
15. Floyd, R.W.: Algorithm 97: shortest path. Communications of the ACM **5**(6), 345 (1962)
16. Fraccascia, L., Yazan, D.M.: The role of online information-sharing platforms on the performance of industrial symbiosis networks. Resources, Conservation and Recycling **136**, 473–485 (2018)
17. Gatzioura, A., Sànchez-Marrè, M., Gibert, K.: A hybrid recommender system to improve circular economy in industrial symbiotic networks. Energies **12**(18), 3546 (2019)

18. Gonela, V., Zhang, J., Osmani, A.: Stochastic optimization of sustainable industrial symbiosis based hybrid generation bioethanol supply chains. Computers & industrial engineering **87**, 40–65 (2015)
19. Kearns, M.: Graphical games. The New Palgrave Dictionary of Economics: Volume 1–8 pp. 2547–2549 (2008)
20. Mas-Colell, A., Whinston, M.D., Green, J.R., et al.: Microeconomic theory, vol. 1. Oxford university press New York (1995)
21. Myerson, R.B.: Graphs and cooperation in games. Mathematics of operations research **2**(3), 225–229 (1977)
22. Roth, A.E., Verrecchia, R.E.: The shapley value as applied to cost allocation: a reinterpretation. Journal of Accounting Research pp. 295–303 (1979)
23. Rubino, R., Omicini, A., Denti, E.: Computational institutions for modelling norm-regulated MAS: An approach based on coordination artifacts. In: Proceedings of AAMAS-2005. pp. 127–141. Springer (2005)
24. Searle, J.R.: What is an institution? Journal of institutional economics **1**(1), 1–22 (2005)
25. Shapley, L.S.: A value for n-person games. Contributions to the Theory of Games **2**(28), 307–317 (1953)
26. Shapley, L.S.: On balanced sets and cores. Naval research logistics quarterly **14**(4), 453–460 (1967)
27. SHAREBOX: SHAREBOX: Secure Sharing. `http://sharebox-project.eu/` (2019), accessed: 2020-10-20
28. Silverman, B.S.: Technological resources and the direction of corporate diversification: Toward an integration of the resource-based view and transaction cost economics. Management science **45**(8), 1109–1124 (1999)
29. Singh, M.P.: Norms as a basis for governing sociotechnical systems. ACM Transactions on Intelligent Systems and Technology (TIST) **5**(1), 21 (2013)
30. Sun, L., Rangarajan, A., Karwan, M.H., Pinto, J.M.: Transportation cost allocation on a fixed route. Computers & Industrial Engineering **83**, 61–73 (2015)
31. Williamson, O.E.: Transaction-cost economics: the governance of contractual relations. The journal of Law and Economics **22**(2), 233–261 (1979)
32. Yazan, D.M., Yazdanpanah, V., Fraccascia, L.: Learning strategic cooperative behavior in industrial symbiosis: A game-theoretic approach integrated with agent-based simulation. Business strategy and the environment (2020)
33. Yazdanpanah, V., Yazan, D.M.: Industrial symbiotic relations as cooperative games. In: Proceedings of the 7th International Conference on Industrial Engineering and Systems Management (IESM-2017). pp. 455–460 (2017)
34. Yazdanpanah, V., Yazan, D.M., Zijm, W.H.M.: Normative industrial symbiotic networks: a position paper. In: Multi-Agent Systems and Agreement Technologies, EUMAS-AT 2016, pp. 314–321. Springer (2016)
35. Yazdanpanah, V., Yazan, D.M., Zijm, W.H.M.: FISOF: A formal industrial symbiosis opportunity filtering method. Engineering Applications of Artificial Intelligence **81**, 247–259 (2019)
36. Yazdanpanah, V., Yazan, D.M., Zijm, W.H.M.: A multiagent framework for coordinating industrial symbiotic networks. In: 21st Workshop "From Objects to Agents" (WOA 2020), 14–16 September 2020. p. to appear (2020)
37. Young, H.P.: Cost allocation: methods, principles, applications. North Holland Publishing Co. (1985)

Swarm Construction Coordinated through the Building Material

Yating Zheng[1,2], Michael Allwright[2], Weixu Zhu[2], Majd Kassawat[3], Zhangang Han[1], and Marco Dorigo[2]

[1] School of Systems Science, Beijing Normal University, Beijing, China
zhengyating@mail.bnu.edu.cn, zhan@bnu.edu.cn
[2] IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
{michael.allwright, weixu.zhu, marco.dorigo}@ulb.ac.be
[3] Universidad Jaume I, Castellon, Spain
majd@uji.es

Abstract. This paper demonstrates a swarm robotics construction system where the intelligence that coordinates construction has been moved from the robots to an advanced building material. This building material is known as the Stigmergic Blocks and is capable of computation and local communication. Using comprehensive simulation models based on real hardware, we investigate approaches to improving the efficiency and flexibility of a swarm robotics construction system.

Keywords: Swarm Robotics · Construction · Simulation.

1 Introduction

In swarm robotics, groups of robots coordinate their actions by communicating with their neighbors and by sensing and modifying the surrounding environment [5,7]. These interactions between the robots and their environment can result in the emergence of useful collective behaviors. It is the goal of swarm robotics researchers to understand how the individual robots in these swarms can be programmed so that these collective behaviors not only perform a useful task but do so in a way that is generalizable, scalable, and robust to disturbances such as robot failures. If these characteristics can be realized in robot swarms, this approach to robotics may be well suited to automating construction in hostile environments. As an example, environments with excessive radiation are too dangerous for human workers and may result in high failure rates of robots and their supporting positioning and communication infrastructure.

From an abstract perspective, the goal of construction is to arrange materials in an environment into one or more structures with respect to a set of constraints. For example, an ordering that ensures that the structure remains stable during the entire building process. In the case of swarm robotics, these constraints can be realized in terms of reactive rules that instruct robots to perform construction actions in response to environmental stimuli. If these stimuli are defined in terms of the results of previous construction actions by other robots, we say

that the robots are coordinating a construction task through stigmergic communication [6,15]. This approach to construction has been applied by Allwright et al. to build a staircase using a single robot and a stepped pyramid using four robots [1,2] and by Jones and Matarić to build 2D structures from colored blocks [8,9]. A significant challenge in this approach, however, is finding a set of rules that unambiguously map all intermediate construction states to construction actions. The complexity of these sets of rules increases with the size of the structure and has necessitated the use of offline algorithms to generate rule sets in similar research [10]. Moreover, if we want to take advantage of the potential scalability of swarm robotics systems by building in parallel, this complexity is exacerbated since building in parallel imposes additional constraints on a rule set to guarantee that the structure is always in a valid state [4,14].

To work around these limitations, researchers have supplemented stigmergic communication in a variety of ways. For example, Werfel et al. [16,18] use the concept of extended stigmergy in their work on multi-robot construction. This approach leverages a robot's or a block's ability to localize itself to simplify the construction rules. The work by Sugawara and Doi [12,13] takes another approach and instead has the building materials guide the robots to where building material should be added. In this paper, we extend the work of Sugawara and Doi by further investigating the potential advantages of having a building material coordinate construction in a more capable multi-robot construction system, namely, the one designed by Allwright et al. [3]. This construction system consists of two components, a robot called the BuilderBot and a building material, called the Stigmergic Block, which the BuilderBot assembles into structures using its manipulator (see Fig. 1). We have developed plugins that provide comprehensive models of the BuilderBot and the Stigmergic Block for the ARGoS simulator [11] and used them in the experimental work presented in this paper.

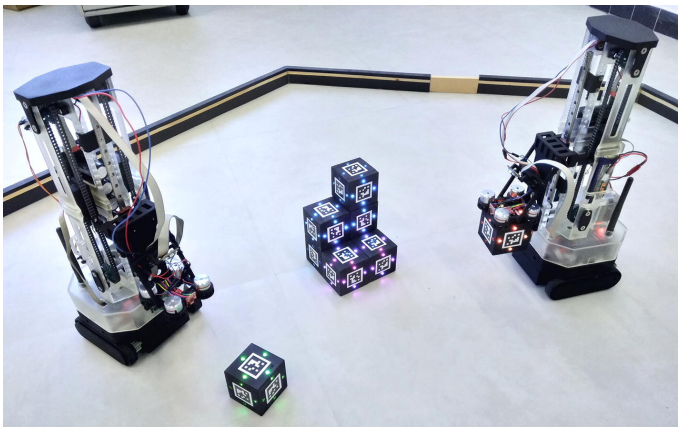


Fig. 1. The Swarm Robotics Construction System (SRoCS) consists of two components, the BuilderBot robot and the Stigmergic Block building material.

The general setup of our construction system involves having the robots use computer vision to identify the configuration of a structure by observing the location of its blocks and the colors of the LEDs on those blocks. The robots then perform construction actions such as attaching another block in response to certain configurations of the structure. In the experiments where we extend the work of Sugawara and Doi, we use the building material's peer-to-peer near-field communication to allow messages to be exchanged between adjacent blocks. By enabling the routing of messages through intermediate blocks, we enable one block to monitor the structure and to communicate directly with the robots by changing the colors of the LEDs on one or more blocks.

The remainder of this paper is organized as follows. In the following section, we describe two classes of construction algorithms that we use to coordinate construction. In Section 3, we present three experiments that demonstrate how the efficiency and flexibility of the building process can be improved and how the need to find complex sets of construction rules can be eliminated by enabling the building material to coordinate its own assembly. Where possible, we compare this approach with a standard approach where the construction is coordinated exclusively by the robots. In Section 4, we discuss the tolerance of our system to faults and the trade-offs that are made by moving the intelligence into the blocks. We conclude the paper in Section 5 by suggesting several directions for future work. The results presented in this paper and the tools required to reproduce those results are open source and available as an OSF project [19].

2 Construction algorithms

In this paper, we use two classes of algorithms for coordinating construction. The first class of algorithms, referred to as the *standard algorithms*, is a generalization of the approach used by Allwright et al. [1] and is used for comparison with the second class of algorithms. This second class of algorithms is called *block algorithms* and represents the approach where the intelligence that coordinates construction has been moved into the building material.

2.1 Standard algorithms

In the standard algorithms, construction is coordinated exclusively through stigmergic communication. The robots perform a random walk in their environment, avoiding obstacles and searching for building material to attach to a structure. The robots perform construction actions as a response to their observations of the results of previous construction actions. In a standard algorithm, the robots are provided with a look-up table that associates intermediate construction states with construction actions. We assume here that the robots do not have access to global information and would not be able to sense the complete state of larger structures. Therefore, an entry in this look-up table often does not contain the entire intermediate construction state, but rather only a partial representation of that state. This partial representation corresponds to a configuration of blocks

that can be reliably detected by a robot's camera. The robots use this look-up table and their sensor readings to detect patterns of blocks in their environment and to execute the construction actions associated with them.

In our experiments with the standard algorithm, we allow robots to change the colors of the LEDs on the Stigmergic Blocks just before attaching them to a structure. Changing the LED colors on a Stigmergic Block enables a BuilderBot to detect more complex patterns of blocks with its computer vision system more reliably. After a BuilderBot has attached a Stigmergic Block to the structure, however, the block's LED colors are fixed.

2.2 Block algorithms

In a block algorithm, the intelligence that coordinates construction is mainly in the building material. Similar to the standard algorithms, the robots perform a random walk in the environment, avoiding obstacles and searching for building material that can be added to an incomplete structure. In a block algorithm, however, the robots do not have any internal representation of the structure being built and rely on the building material for coordination.

In our system, construction starts with a single *root* Stigmergic Block in the environment. While in our experiments we assign the role of the root block statically, it would also be possible to have one or more robots assign this role to one or more blocks dynamically as a result of environmental stimuli. The root block in our current implementation of a block algorithm contains the entire target structure encoded as a rooted tree. The root block decomposes this rooted tree and sends only the required branches to its children using peer-to-peer near-field communication (NFC). This process continues until all blocks currently in the structure have received instructions from the root block. The non-root blocks in the structure continuously send data back to their parents who then forward the received data back to their parents as a single message until the root block has been reached.

Upon receiving the messages from its children, the root block can monitor construction progress, can detect incorrectly placed blocks, and can update the colors of the LEDs on the Stigmergic Blocks in the structure, triggering further construction actions by the BuilderBots. By controlling these LEDs, the root block is able to coordinate the construction of the structure by telling nearby robots where further blocks can be attached or should be removed.

Although this paper focuses primarily on results from simulation, we have successfully implemented a block algorithm using the Stigmergic Blocks, whose hardware is described in [3]. A video of this algorithm working on the hardware (with blocks being attached and detached by hand) is available online as part of the OSF project.[4] In the following section, we describe our experiments in simulation.

[4] Video: `hardware-demo.mp4` at https://osf.io/ve3za/

3 Experiments

In this section, we present three experiments that we have completed using the models of the BuilderBot and Stigmergic Block in the ARGoS simulator. We model the behavior of the Stigmergic Block firmware in ARGoS using a Lua controller that allows callbacks to be executed while messages are being exchanged. This model reflects the actual hardware with the exception that the firmware for the real block is written in C++ and is interrupt-driven, while the code used in simulation is written in Lua and uses polling to detect if a neighboring block is attempting to exchange messages. The control software for the BuilderBot robot is also written in Lua and uses a behavior tree architecture. An API for the BuilderBot has been developed, which provides a library of behavior trees for obstacle avoidance, picking up unused blocks, and attaching them to structures following rules that have been defined in terms of patterns of blocks that can be detected by the robot's computer vision system.

Our first experiment demonstrates a concept called dynamic construction paths, where the root block is able to adjust the target structure as it is being built. In the second experiment, we show how the blocks can be used to guide a robot towards a vacant construction site. Finally, the third experiment demonstrates how using a block algorithm allows for a more flexible construction process where robots can attach blocks to any vacant construction site in any order. These experiments aim to demonstrate the potential advantages of moving the intelligence that coordinates construction from the robots to the blocks.

3.1 Dynamic construction paths

In the standard algorithms, the Stigmergic Blocks are unable to communicate with each other, they can only have their LEDs configured by a robot to display a certain color before they are attached to a structure. The robots change the color of the blocks as part of executing a construction action. The set of rules that maps the intermediate construction states to these construction actions is prepared offline and is loaded into the memory of the robots before an experiment is started. In contrast, the block algorithms only require the root block to have the internal representation of the structure, which can also be modified during construction. This capability enables a feature called dynamic construction paths. The concept of dynamic construction paths is realized when two or more sequences of construction actions can be selected during construction according to a condition that can be detected by the root block (or one of the blocks with which it is in communication).

In this section, we set up an experiment with a structure that can be completed by following one of four different construction paths. This structure is shown in Fig. 2. If we ignore the orientation of the structure, there are four construction paths that advance the state of the structure from what is shown in Fig. 2a to Fig. 2c. That is, we can attach blocks (i) left and then right, (ii) right and then left, (iii) front and then back, or (iv) back and then front.

In this experiment, the root block decides which path to follow by initially indicating that a block can be attached to the top face of either the left, right, front, or back block (Fig. 2a). Once a block has been attached to one of these sites (and this information has propagated back to the root block), the root block updates the illumination pattern of the structure to show nearby robots that there is one valid construction site remaining (Fig. 2b). Following the attachment of a block to this site, the root block updates the illumination pattern of the structure one last time to indicate to nearby robots that the structure is complete (Fig. 2c).

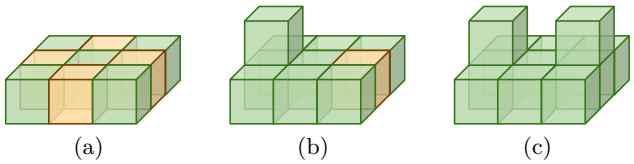


Fig. 2. Structure for demonstrating the use of a block algorithm to dynamically select construction paths. The root block in the structure indicates to nearby robots when and where a block can be attached to the structure by setting the color of a valid construction site to yellow.

Results from simulation. The image on the left of Fig. 3 shows a robot approaching the partially built structure. At this point, all four construction paths are possible. After the robot has placed the block on the right-hand side of the structure, the root block disables the LEDs on the right, front, and back blocks to indicate that a block can now only be added on the left. The structure is completed when the robot adds this last block to the structure, as shown on the right of Fig. 3.

To investigate the impact of using more robots, we repeated this experiment with two and four robots. Each of these configurations was repeated 25 times, with the blocks and the robots starting in random positions. Each experiment was automatically terminated when all required blocks have been deposited at the building sites. The videos and the source code for these experiments are available as part of the OSF project for this paper.[5]

Fig. 4 shows the distribution of the total experiment time with one, two, and four BuilderBots. While there is a decrease in the time taken between one and

[5] Videos: `dcp-single-robot.mp4` and `dcp-multiple-robots.mp4` at https://osf.io/9562j/ and https://osf.io/4cpyh/
Source code: `dcp-single-robot.zip` and `dcp-multiple-robots.zip` at https://osf.io/j2pqh/ and https://osf.io/nasf6/

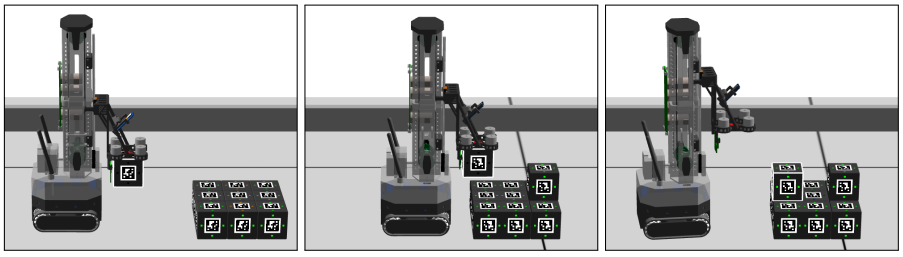


Fig. 3. Simulation results for the dynamic construction paths experiment. From left to right: (i) a robot approaches a partially built structure and places a block on top of one of the orange faces, (ii) the root block responds by selecting a construction path, changing the illumination pattern, (iii) the robot places the final block in the correct location to complete the structure.

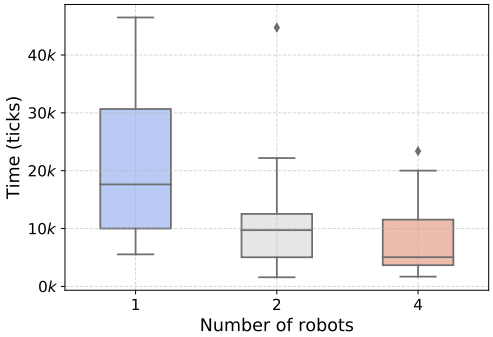


Fig. 4. Distribution of simulation time with different number of BuilderBots.

two robots, the decrease between two and four robots is less significant. This diminishing return on increasing the number of robots is commonly observed in swarm robotics systems since adding more robots to a system increases the likelihood of interference between those robots. From this data, we may conclude that a swarm of two robots is optimal for this particular construction task in this environment.

3.2 Guided construction

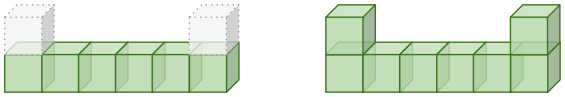
In this experiment, we show how a block algorithm can also be used to guide a robot towards a construction site. This configuration involves using the illumination pattern on the blocks to communicate the direction in which a robot should go to reach a construction site. The motivation behind implementing this mechanism is that, in the standard algorithms, the robot tends to spend a lot of time performing a random walk before locating a construction site where it can attach a block. The idea of using the building material to guide robots towards a construction site is sometimes referred to as *gradient following* and has been demonstrated before in a more abstract simulation by Werfel et al. [17].

For example, consider the partially built structure consisting of six blocks arranged in a line in Fig. 5a. To complete this structure, a robot must place one block on top of the left block and one block on top of the right block (Fig. 5b). However, since the perspective of the robot is limited, it must discover these attachment sites either through random walk or through gradient following.

In a standard algorithm, the colors of the LEDs on the blocks can not be updated once they have been attached to a structure. For this reason, the robot must rely on random walk to discover the possible attachment sites. Fig. 6 shows how this construction may take place. The robots' rule set in this case is that a green block is to be attached to the top of a yellow block (unless a green block has already been attached).

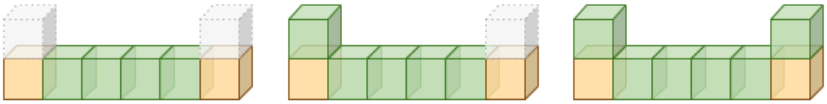
The construction speed for this structure can be increased using a block algorithm that implements gradient following. In this case, the illumination pattern of the structure is under the control of the root block and can be updated in response to changes in the structure. Moreover, the robots now follow three rules: (i) when a yellow block is detected the robot attaches a block to the top of it, (ii) if red blocks are detected the robot biases its random walk behavior to the right, (iii) if blue blocks are detected, the robot biases its random walk behavior to the left. Fig. 7 shows an example of how this construction may take place. In this example, a block is attached on top of the leftmost block, which is detected by the root block. The root block updates the illumination pattern so that a robot approaching the structure will turn to the right and find the remaining construction site.

Results from simulation. To test our hypothesis that guided construction with a block algorithm reduces the overall construction time with respect to what is possible with a standard algorithm, we run experiments with two structures: a short line composed of six blocks and a long line made up of thirteen



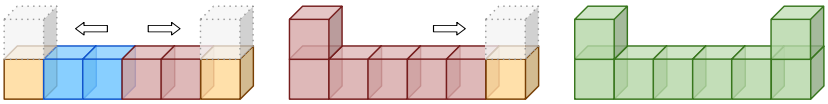
(a) (b)

Fig. 5. Structure for demonstrating guided construction. (a) The initial state of the structure is a line consisting of six blocks. (b) The structure is completed by placing a block at each end of the structure.



(a) (b) (c)

Fig. 6. Construction of the structure in Fig. 5 using a standard algorithm.



(a) (b) (c)

Fig. 7. Construction of the structure in Fig. 5 using a block algorithm to indicate which way a robot should turn to reach a valid construction site.

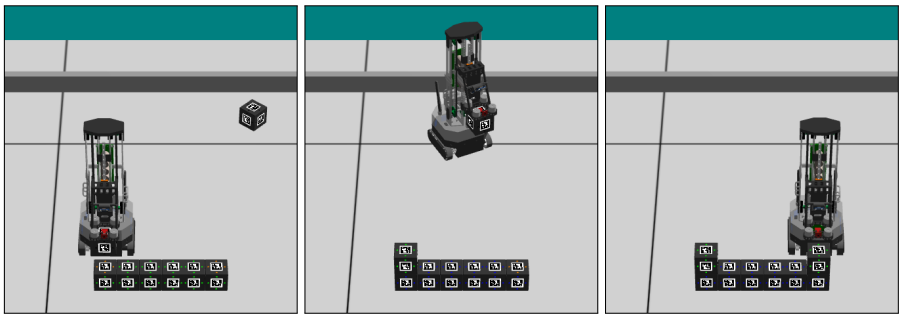


Fig. 8. Simulation of building the structure in Fig. 5 with a block algorithm in the ARGoS simulator. (a) The robot attaches a block to the top of the leftmost block. (b) The illumination pattern is updated by the root block and robot searches to the right for possible construction sites. (c) The robot attaches the last block to the top of the rightmost block.

blocks. We run each experiment for the two structures 25 times using both the standard algorithm and the block algorithm. Fig. 8 contains three screenshots of the construction of the short structure with a block algorithm in the ARGoS simulator.

The box plot in Fig. 9 shows that, for both structures, the approach based on the block algorithm performed better than the approach based on the standard algorithm. From comparing the results for the two structures, it appears that the decrease in construction time is related to the size of the structure, however, further experiments with different types of structures and varying numbers of robots are needed to get a proper insight into this relationship. The videos and the source code for reproducing these experiments are available as part of the OSF project.[6]

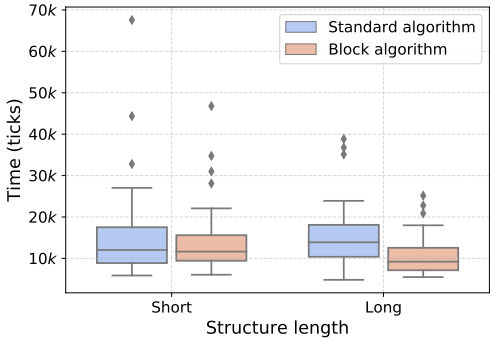


Fig. 9. Distribution of the time taken to construct the short and long structures using a standard algorithm and a block algorithm.

3.3 Flexible construction

Implementing construction in a swarm robotics system using a standard algorithm puts a heavy burden on the designer to come up with a set of rules that unambiguously maps each intermediate state of a structure to a construction action. This burden is only made worse when we want to design rules that facilitate flexible construction. For example, consider the structure in Fig. 10. If we wanted to build this structure using the standard algorithm, we could constrain the building process so that there is only one construction path that can be followed, that is, there is exactly one construction action associated with each intermediate state (Fig. 11). This constrained approach, however, may be

[6] Videos: `gc-standard-algorithm.mp4` and `gc-block-algorithm.mp4` at https://osf.io/5h9cs/ and https://osf.io/cdvty/
Source code: `gc-standard-algorithm.zip` and `gc-block-algorithm.zip` at https://osf.io/we754/ and https://osf.io/3znua/

inefficient, since a robot could approach a possible construction site but be prohibited to attach a block due to the constraints of the rule set. In contrast to the constrained approach, if we allow a building process where a robot can attach a block to any possible construction site at any time, the number of possible intermediate states would increase significantly. Even for the simple structure in Fig. 10, the number of intermediate states increases from three to seven. Finding the unambiguous mappings between all of these intermediate states and the possible construction actions that advance the building process while keeping the structure in a valid state is at least difficult and may in many cases be infeasible.

A block algorithm can solve this problem since the root block can detect when and where one or more blocks have been added to (or removed from) a structure and can update the illumination pattern on the blocks accordingly. Furthermore, in the case of a block being attached to an incorrect site, the root block can detect the incorrectly placed block and update the illumination pattern so that nearby robots remove it, restoring the structure to a valid intermediate state. In the final experiment for this paper, we demonstrate the construction of the structure in Fig. 10 using the ARGoS simulator.

Fig. 10. Candidate structure for flexible construction. (a) initial state of the structure, (b) target state of the structure.

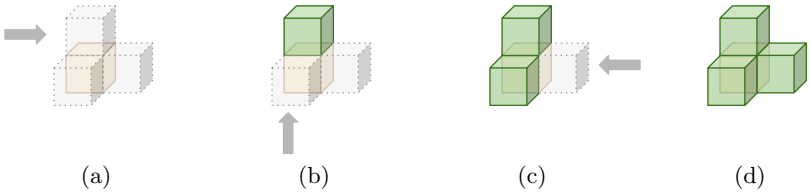


Fig. 11. The structure in Fig. 10, can be built sequentially using a standard algorithm to map the possible intermediate states of the structure (configurations of blocks) to construction actions.

Results from simulation. We have implemented the construction of the structure in Fig. 10 using a standard algorithm for sequential construction with a single robot (Fig. 12) and with a block algorithm for construction of the same structure with three robots in parallel. Videos of these experiments are available online as part of our OSF project for this research.[7]

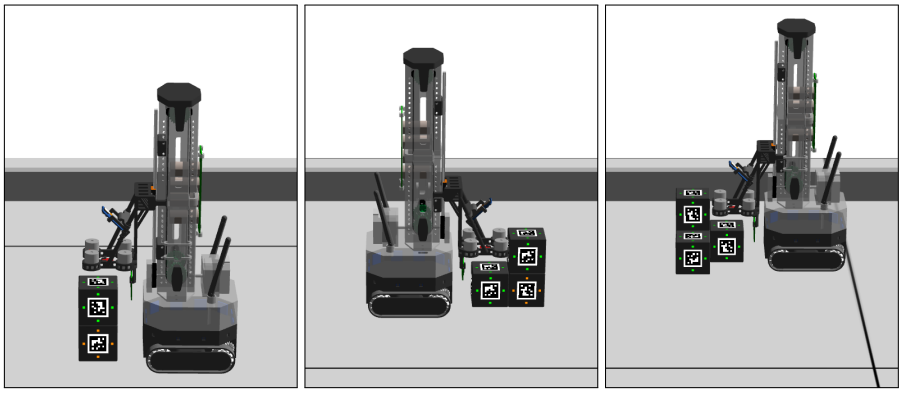


Fig. 12. Construction of the structure in Fig. 10 using a standard algorithm with a single robot

4 Discussion

4.1 Fault Tolerance

In this section, we discuss two types of faults and how the standard and block algorithms can recover from them. The first type of fault is when a robot attaches a block to an incorrect site. This fault can be caused by a sensor error on the behalf of the robot or can be due to unfortunate timing. For example, when two or more robots attach blocks to valid attachment sites but where the combination of those attachments puts the structure into an incorrect state. The second type of fault is when a block stops working correctly. This fault may be the result of a bad power source, corrupted firmware, or damaged hardware.

The standard algorithms can handle the first type of fault, where a block has been incorrectly attached to a structure, at the cost of increasing the complexity of the ruleset. That is, in addition to the rules necessary to advance the construction, it would be possible to add rules that match the structure when

[7] Videos: `fc-standard-algorithm.mp4` and `fc-block-algorithm.mp4` at https://osf.io/ycxes/ and https://osf.io/tvhs2/
Source code: `fc-standard-algorithm.zip` and `fc-block-algorithm.zip` at https://osf.io/gf94r/ and https://osf.io/kjhu7/

it is in an incorrect state and that trigger the removal of one or more blocks until the structure is back in a state from which the construction can continue. The second type of fault is difficult to solve with the standard algorithm and relies on the robots being able to infer that a block is faulty, e.g., the LEDs are displaying the wrong color. If the robots detect a faulty block in the structure, it can be ignored or removed if it is disruptive to the building process.

For the block algorithms, the first type of fault, where a block has been attached to an incorrect site, can be resolved since the root block can detect the presence of this block by exchanging messages with other blocks in the structure and can update the illumination pattern of the structure so that the robots remove it. A demonstration of a block algorithm recovering from this fault has been implemented for the dynamic construction paths discussed in Section 3.1. A video of the recovery from this fault is available as part of the OSF project along with the source code to reproduce the experiment[8].

The second type of failure, that is, if the block has (i) a bad power source, (ii) corrupted firmware, or (iii) damaged hardware, is more problematic for block algorithms than standard algorithms since the block algorithms currently rely on the accurate propagation of information through the structure. In some cases, it may be possible to work around these malfunctioning blocks by communicating through other blocks, however, thin sections of the structure where there is only a single path through which information can flow remain problematic and will require further research.

4.2 Trade-offs

Although the experiments in this paper show that the block algorithms can make construction more flexible and efficient and can put less of a burden on the system designer, there are some important trade-offs that must be addressed. The first trade-off is the increase in complexity of the building materials, which can no longer be passive but now have to be capable of computation and local communication, which increases the cost and necessitates a source of power. This trade-off, however, is not so unreasonable considering recent developments in smart label technology where NFC communication, small micro-controllers, and lithium batteries can be combined into cheap flexible tags that could be attached to building materials in an automated construction system.

The second trade-off that must be considered is that a block algorithm uses a root block in the structure to coordinate its construction, introducing a form of centralized control which may be undesirable since (i) it is a potential bottleneck in terms of computational and communication throughput and (ii) it creates a single point of failure in the system. We believe, however, that it is feasible to use centralized control in a swarm robotics construction system without negating the benefits of decentralized control as long as the following conditions can be met: (i) the role of the centralized controller can be transferred to another unit in the

[8] Video: `dcp-fault-tolerance.mp4` at https://osf.io/mvhk6/
Source code: `dcp-fault-tolerance.zip` at https://osf.io/scm7q/

case of hardware failure and (ii) the centralized controller can partially delegate its authority to other units so that it is not a computational/communication bottleneck in the system.

5 Conclusion

In this paper, we demonstrated the advantages of moving the intelligence that coordinates a building process in a swarm robotics construction system from the robots and into the building material. We referred to these algorithms as block algorithms and compared them against solutions where the intelligence that coordinates construction was in the robots, namely the standard algorithms.

In future work, we intend to investigate the scalability and fault tolerance of the block algorithms and to validate the experiments presented in this paper using real robots.

Acknowledgements

This work is partially supported by the Program of Concerted Research Actions (ARC) of the Université libre de Bruxelles, by a Research Credit (CDR -Crédit de Recherche) grant from the Belgian F.R.S.-FNRS, and by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 846009. Yating Zheng and Weixu Zhu would like to acknowledge their support from the China Scholarship Council (grant numbers 201806040106 and 201706270186). The research in this paper was partly undertaken at the UJI Robotic Intelligence Laboratory. Support for this laboratory is provided in part by Ministerio de Economía y Competitividad (DPI2015-69041-R) and by Universitat Jaume I (UJI-B2018-74). Marco Dorigo acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Director.

References

1. Allwright, M., Bhalla, N., Dorigo, M.: Structure and markings as stimuli for autonomous construction. In: 2017 18th International Conference on Advanced Robotics (ICAR). pp. 296–302. IEEE (2017)
2. Allwright, M., Bhalla, N., Pinciroli, C., Dorigo, M.: Simulating multi-robot construction in argos. In: International Conference on Swarm Intelligence. pp. 188–200. Springer (2018)
3. Allwright, M., Zhu, W., Dorigo, M.: An open-source multi-robot construction system. HardwareX **5**, e00050 (2019)
4. Bonabeau, E., Guérin, S., Snyers, D., Kuntz, P., Theraulaz, G.: Three-dimensional architectures grown by simple 'stigmergic' agents. BioSystems **56**(1), 13–32 (2000)
5. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: A review from the swarm engineering perspective. Swarm Intelligence **7**(1), 1–41 (2013)

6. Grassé, P.P.: La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. Insectes sociaux **6**(1), 41–80 (1959)
7. Hamann, H.: Swarm robotics: A formal approach. Springer (2018)
8. Jones, C., Matarić, M.J.: From local to global behavior in intelligent self-assembly. In: 2003 IEEE International Conference on Robotics and Automation. pp. 721–726. IEEE (2002)
9. Jones, C., Matarić, M.J.: Automatic synthesis of communication-based coordinated multi-robot systems. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 381–387. IEEE (2004)
10. Petersen, K.H., Nagpal, R., Werfel, J.K.: Termes: An autonomous robotic system for three-dimensional collective construction. Robotics: science and systems VII (2011)
11. Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L.M., Dorigo, M.: ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence **6**(4), 271–295 (2012)
12. Sugawara, K., Doi, Y.: Collective construction of dynamic structure initiated by semi-active blocks. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 428–433. IEEE (2015)
13. Sugawara, K., Doi, Y.: Collective construction by cooperation of simple robots and intelligent blocks. In: International Conference on Intelligent Robotics and Applications. pp. 452–461. Springer (2016)
14. Theraulaz, G., Bonabeau, E.: Coordination in distributed building. Science **269**(5224), 686–688 (1995)
15. Theraulaz, G., Bonabeau, E.: A brief history of stigmergy. Artificial life **5**(2), 97–116 (1999)
16. Werfel, J., Nagpal, R.: Extended stigmergy in collective construction. IEEE Intelligent Systems **21**(2), 20–28 (2006)
17. Werfel, J., Nagpal, R.: Three-dimensional construction with mobile robots and modular blocks. The International Journal of Robotics Research **27**(3-4), 463–479 (2008)
18. Werfel, J., Petersen, K., Nagpal, R.: Designing collective behavior in a termite-inspired robot construction team. Science **343**(6172), 754–758 (2014)
19. Zheng, Y., Allwright, M., Weixu, Z., Majd, K., Han, Z., Dorigo, M.: Hybrid coordination for swarm construction. OSF (2020)

Compressed contributions

State Aggregation and Deep Reinforcement Learning for Knapsack Problem

Reza Refaei Afshar Yingqian Zhang, Murat Firat, and Uzay Kaymak

Eindhoven University of Technology, Eindhoven, Netherlands

1 Introduction

In [1], we develop a state aggregation method for solving knapsack problems (KP) with deep reinforcement learning (DRL). Although handcrafted heuristics work well in many COPs, they mostly rely on the nature of problems and they need to be revised for different problem statements. In this paper, we aim to learn and improve the handcrafted heuristics to improve the quality of the solutions. We study *knapsack problem (KP)*, and we propose a state aggregation method to shrink state space in order to solve larger KP instances. A tabular RL method is used to learn the best aggregation strategy for each item. This aggregated features reduces the state space by reducing the number of unique values. Then, *Advantage Actor Critic (A2C)* algorithm as a powerful method of Deep Reinforcement Learning (DRL) is employed to learn the policy of selecting items. The proposed method solves KP by successive item selections and placing them in the knapsack, each is done by following a greedy or softmax algorithm on the output of the policy network. The experimental results show that the method obtains close to optimal solutions for three different types of instances with up to 500 items

2 Proposed method

Figure 1 shows the overview of our method. It consists of two components. Algorithm 1 includes a formulation of KP to MDP, which is solved using a DRL approach. Algorithm 2 is a state aggregation method, which learns an aggregation policy to discretize states that serve as inputs to DRL.

DRL knapsack Solver: In order to solve the 0-1 KP, DRL is used to derive a policy through that the items are sequentially added to the solution. The states, actions and rewards of DRL modeling are as follows. *States* $s(P)$*:* A complete set of information of an instance containing the values and the weights of items and capacity of knapsack. *Actions:* There are N

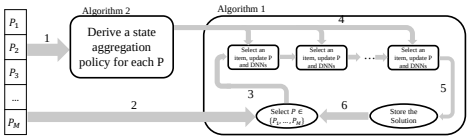


Fig. 1: The overview of the KP solver.

Table 1: Results of different algorithms and datasets of $M = 1000$ instances.

Dataset	Method	N	$\overline{Val}$	$\#_{opt}$	$\overline{Val}_{opt}$
FI	Greedy	500	111.68	204	
	DRL w/o aggregation	500	111.63	64	111.73
	DRL w/ aggregation	500	**111.70**	**261**	
HI	Greedy	500	80779.23	25	
	DRL w/o aggregation	500	81022.60	71	81103.99
	DRL w/ aggregation	500	**81064.99**	**136**	

actions, each corresponding to select one item. *Reward Function:* The reward function contains three terms: a positive reward for successfully selecting an item; A large negative reward when the item does not exist (in case when the number of items is lower than the selected item id); and a small negative number when an item is heavier than the remaining capacity of knapsack. Employing these definitions of states, actions and rewards, the A2C algorithm is used for training policy and value DNNs.

State Aggregation: As the number of items increases, the state space grows up exponentially and this affects the performance of function approximation with DNN. In order to shrink the state space and boost the method to have the capability of solving large problem instances, a new state embedding is derived by state aggregation. Specifically, the problem is to find a certain number of split points on the values of items and transform the values into integers using these split points. We opt for reinforcement learning to tackle this problem and Q-Learning is used to find the optimal number of split points for each item value.

3 Experiments

The proposed DRL with aggregation algorithm is compared with (1) greedy algorithm, (2) DRL without aggregation (3) DRL approach with pointer network (4) Pointer Network and Supervised learning method. We use three different types of instances in the experiments: *Random Instances (RI)*, *Fixed capacity Instances (FI)* and *Hard Instances (HI)*. The DNNs consist of two layers of 64 nodes. We evaluate the performance based on several metrics. The Average Value of Solutions ($\overline{Val}$) and Number of optimally solved instances ($\#_{opt}$) for FI and HI are shown in table 1. The results show that the proposed methods, with or without aggregation, outperform the greedy algorithm. As shown in table 1, the state aggregation strategy improves the solutions of greedy algorithm for large instances. For FI instances, our method finds close to optimal solutions for instances up to 500 items. In the literature, the authors did not test instances with more than 200 items.

References

1. Refaei Afshar, Reza, Yingqian Zhang, Murat Firat, and Uzay Kaymak. 2020. "A State Aggregation Approach for Solving Knapsack Problem with Deep Reinforcement Learning." In Proceedings of the 12th Asian Conference on Machine Learning.

CONLON: A Pseudo-Song Generator Based on a New Pianoroll, Wasserstein Autoencoders, and Optimal Interpolations

Luca Angioloni[1], Tijn Borghuis[2,3], Lorenzo Brusci[3], and Paolo Frasconi[1]

[1] DINFO, Università di Firenze, Italy
first.last@unifi.it
[2] Eindhoven University of Technology, The Netherlands
v.a.j.borghuis@tue.nl
[3] Music-co, Eindhoven, The Netherlands
lorenzo.brusci@musi-co.com

Significant progress in algorithmic music generation has recently resulted from the widespread application of new and powerful methods based on deep generative models, letting this class of data-driven approaches gradually take over more traditional rule-based or probabilistic techniques. The musical quality of the results is still not always sufficient to enable a widespread adoption in realistic professional scenarios. The generation system we present in [1], introduces novelties across three dimensions: the type of data structures that are used to describe MIDI patterns, the nature of the generative learning models, and the strategy used to produce a whole musical piece, whose combination allows us to generate meaningful and professionally usable streams of music. We call our system CONLON, for Channeled Onset of Notes and Length Of Notes, and in honor of Conlon Nancarrow (1912–1997), a pioneer of piano roll compositions.

We introduce a novel pianoroll-like pattern description, PR^{C} , that stores velocities and durations in two separate channels. Our description does not suffer the ambiguity between long notes and repeated occurrences of the same note that is inherent in binary piano roll descriptions (PR). PR^{C} is completely lossless: a quantized MIDI pattern transformed into the corresponding PR^{C} tensor can be recovered exactly. Additionally, it can be perceptually more robust to reconstruction errors. A further advantage is that all the information about a note is local, whereas in the case of PR, a convolutional network requires a wide receptive field to infer the note duration.

As a generative model, we experiment with Wasserstein autoencoders (WAE) [4], a type of autoencoder that is less subject to the "blurriness" problem typically associated with variational autoencoders (VAE), which manifests itself in the case of music patterns as large clusters of notes being played together and sometimes in swarms of short notes that are never present in the training data. WAEs avoid this problem by pushing the expectation inside the divergence, i.e., penalizing a divergence $\mathcal{D}$ between the prior q_z and the *aggregated* posterior $q_z(z) = \mathbb{E}_p\, q(z|x)$, where p is the data distribution. They thus minimize, with respect to the parameters of the decoder, the quantity

$$\min_{q(z|x)} \mathbb{E}_p\, \mathbb{E}_{q(z|x)}\, c(x, G(z)) + \lambda \mathcal{D}(q_z, p_z) \tag{1}$$

where c is a reconstruction loss and λ a hyperparameter to be fixed. In all our experiments we employed the Maximum Mean Discrepancy (MMD) for $\mathcal{D}$ and a Gaussian prior for p_q, and we structured the encoder and the decoder as in the DCGAN [3] architecture.

Our generation strategy is similar to interpolation, where MIDI pseudo-songs are obtained by concatenating patterns decoded from smooth trajectories in the embedding space, but we formulate it as an optimization problem for exploring the autoencoder latent space in a way that prevents abrupt transitions between consecutively generated patterns, as well as regions with little variation. The optimal trajectories are computed as the solution of a widest-path problem.

We tested CONLON on three datasets. ASF-4 is a set of 910 patterns of four bars in three genres: *acid jazz*, *soul* and *funk*. Each pattern has 4 tracks associated with a simple electro-acoustic quartet: drums, bass, Rhodes piano, and Hammond organ. HP-10 is a set of 968 patterns of four bars in two genres: *high-pop* and *progressive trance*. Each pattern has 10 tracks associated with the following instrument set: drums, bass, Rhodes, brass-synth, choir, dark-pad, guitar, lead, pad, and strings. Both ASF-4 and HP-10 have been especially composed by two professional musicians for this study[4]. The third dataset was LPD-5 (cleansed version) derived from the Lakh MIDI dataset by Dong *et al.* [2].

To validate the CONLON approach, we conducted three listening experiments with a group of 69 musicians. These experiments showed that musicians find pseudo-songs generated with WAEs and PR^{C} descriptions more useable in music production than pseudo-songs generated with the MuseGAN model [2] and PR descriptions, find pseudo-songs generated by WAEs with PR^{C} descriptions more useable than pseudo-songs generated by the same WAE with PR descriptions, and find the development over time of pseudo-songs generated with WAEs and PR^{C} description coherent rather than incoherent (with respect to Harmony, Rhythm, Melody, and Interplay of instruments).

References

1. Angioloni, L., Borghuis, T., Brusci, L., Frasconi, P.: Conlon: A pseudo-song generator based on a new pianoroll, wasserstein autoencoders, and optimal interpolations. In: Proceedings of the 21th International Society for Music Information Retrieval Conference ISMIR MTL2020. pp. 876–883 (2020)
2. Dong, H., Hsiao, W., Yang, L., Yang, Y.: Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. pp. 34–41 (2018)
3. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: 4th International Conference on Learning Representations (2016)
4. Tolstikhin, I.O., Bousquet, O., Gelly, S., Schölkopf, B.: Wasserstein auto-encoders. In: 6th International Conference on Learning Representations (2018)

[4] These datasets, along with other additional materials are available at https://paolo-f.github.io/CONLON/.

AI-Toolbox: A Framework for Fundamental Reinforcement Learning*

Eugenio Bargiacchi[1], Diederik M. Roijers[1,2], and Ann Nowé[1]

[1] Department of Computer Science, Vrije Universiteit Brussel
[2] Microsystems Technology, HU University of Applied Sciences
svalorzen@gmail.com

As is generally true for computer science, reinforcement learning (RL) curricula are mainly composed of two parts: theory, and hands on experience. The latter, in particular, often requires the students to implement the most fundamental algorithms from scratch. Students can then test these on toy examples in order to gain an intuitive understanding of the underlying mechanisms, e.g. value functions, belief updates, etc. While useful, this approach can be time-consuming, which prevents students from experimenting with less known or harder to implement methods and comparing their characteristics directly. To counteract this, providing a suite of already implemented methods can help to significantly expand the experience of a novice.

With the recent surge of interest in the field of deep learning, there are numerous resources for implementing and testing deep methods today. However, we argue that deep methods are not optimal for (human) learning, as it is difficult to inspect their internals and get an intuitive understanding of what is happening under the hood. Unfortunately, relatively few libraries organize discrete literature that can be used for this purpose [7, 6, 4].

AI-Toolbox[3] is a C++\Python library that tries to fill this gap. It is a research-grade repository of more than 40 implementations of several algorithms for single-agent and multi-agent bandit, MDP and POMDP algorithms, and also provides a large number of utility classes and functions available to implement additional methods. AI-Toolbox is one of the largest frameworks of its type available online, and it is free software.

AI-Toolbox is battle-tested, and has been referenced by numerous publications [9, 2, 8, 5, 1]. The goals of this framework are, in descending order of importance: usability and documentation, ease of modification, clarity and performance. These goals align well with student experimentation and discovery.

AI-Toolbox extensive documentation covers every public class, method and utility. The library provides a uniform, consistent interface throughout, emphasizing patterns across algorithms that might not otherwise be noticed. The code is written in C++17, taking advantage of all features of the language, and is built following modern standard practices, i.e. unit tests, continuous integration, separate concerns, etc.

To give a brief example of how easy it is to use the library, this is all the code needed to use the Incremental Pruning algorithm to solve the known POMDP

* Supported by FWO (Fonds Wetenschappelijk Onderzoek), grant #1SA2820N, and by the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

tiger problem, and initialize a policy with the resulting value function. The policy can then be easily used to act in the environment; for example, to control a robot.

```
// The model can be any custom class that respects a 10-method interface.
// In this case it is a problem provided by the toolbox.
auto model = AIToolbox::POMDP::makeTigerProblem();
unsigned horizon = 10; // The horizon of the solution.

// The 0.0 is the convergence parameter. It gives a way to stop the
// computation if the policy has converged before the horizon.
AIToolbox::POMDP::IncrementalPruning solver(horizon, 0.0);

// Solve the model and obtain the optimal value function.
auto [bound, valueFunction] = solver(model);

// We create a policy from the solution to compute the agent's actions.
// The parameters are the size of the model (SxAxO), and the value function
// obtained from solving the problem.
AIToolbox::POMDP::Policy policy(2, 3, 2, valueFunction);
```

References

1. Arming, S., Bartocci, E., Chatterjee, K., Katoen, J.P., Sokolova, A.: Parameter-independent strategies for pMDPs via POMDPs. In: International Conference on Quantitative Evaluation of Systems. pp. 53–70. Springer (2018)
2. Bargiacchi, E., Verstraeten, T., Roijers, D., Nowé, A., van Hasselt, H.: Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In: International Conference on Machine Learning. vol. 80, pp. 482–490. PMLR (10–15 Jul 2018)
3. Bargiacchi, E., Roijers, D.M., Nowé, A.: AI-Toolbox: A C++ library for reinforcement learning and planning (with Python bindings). Journal of Machine Learning Research **21**(102), 1–12 (2020), http://jmlr.org/papers/v21/18-402.html
4. Chades, I., Chapron, G., Cros, M., Garcia, F., Sabbadin, R.: MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. In: Ecography. vol. 37, pp. 916–920 (2014)
5. Chatterjee, K., Elgyütt, A., Novotný, P., Rouillé, O.: Expectation optimization with probabilistic guarantees in pomdps with discounted-sum objectives. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 4692–4699. International Joint Conferences on Artificial Intelligence Organization (7 2018). https://doi.org/10.24963/ijcai.2018/652, https://doi.org/10.24963/ijcai.2018/652
6. Egorov, M., Sunberg, Z., Balaban, E., Wheeler, T., Gupta, J., Kochenderfer, M.: POMDPs.jl: A framework for sequential decision making under uncertainty. Journal of Machine Learning Research **18**(26), 1–5 (2017), http://jmlr.org/papers/v18/16-300.html
7. Oliehoek, F., Spaan, M.T.J., Terwijn, B., Robbel, P., Messias, J.V.: The MADP toolbox: An open-source library for planning and learning in (multi-)agent systems. Journal of Machine Learning Research **18**(1), 3112–3116 (2017)
8. Petrák, J.: Maze Navigation via Monte Carlo Tree Search. Ph.D. thesis, Masaryk University (2019)
9. Verstraeten, T., Bargiacchi, E., Libin, P., Helsen, J., Roijers, D.M., Nowé, A.: Multi-agent Thompson sampling for bandit applications with sparse neighbourhood structures. Nature Scientific Reports **10**(1), 6728 (2020)

Extraction of high-level features and labels in multi-label classification problems

Marilyn Bello[1,2], Gonzalo Nápoles[2,3], Ricardo Sánchez[1], Koen Vanhoof[2], and Rafael Bello[1]

[1] Computer Science Department, Central University of Las Villas, Cuba
[2] Faculty of Business Economics, Hasselt University, Belgium
[3] Department of Cognitive Science & Artificial Intelligence, Tilburg University, The Netherlands
mbgarcia@uclv.cu

1 Introduction

Deep learning [4] is a promising avenue of research into the automated extraction of complex data representations at high levels of abstraction. Such algorithms develop layered, hierarchical learning architectures of data representations, where higher-level features are defined in terms of lower-level features. Pooling layers [4, 6] provide an approach to downsampling feature maps by summarizing the presence of features in patches of the feature map. They focus on data with a well-defined structure where the term feature neighborhood makes sense.

However, while it is interesting to recognize faces, or classify objects in images and videos, the truth is that there are other domains in which the data do not have a topological organization [7, 8]. For example, when using numerical descriptors to encode a protein, it might happen that two distant positions in the sequence are close to each other in the tri-dimensional space. That behavior could not be captured with local pooling methods. In those cases, using standard pooling operators might have little sense, even when the problem at hand could benefit significantly from a deep learning solution. Besides, although these operators are able to deal with both single-label and multi-label classification (MLC) problems [3], they are specifically aimed at reducing feature space. However, in the case of multi-label data, we can benefit significantly from implementing similar operations on the label space.

Hence, we propose a deep neural architecture to extract high-level features and labels in MLC problems [1, 2]. This approach, unlike the classic use of pooling, does not pool pixels but problem features or labels. The following sections provide a brief description of our proposal.

2 Bidirectional Deep Neural Network

This architecture, proposed in [1], is composed of several stacked association-based pooling layers, which are built starting from the features and the labels at the same time. The first pooling layer is composed of neurons denoting the

problem features and labels, whereas in deeper pooling layers the neurons denote high-level features and labels extracted during the construction process. Each pooling layer uses a function that detects pairs of highly associated neurons (i.e. they fulfill a certain association threshold) while performing an aggregation operation to derive the pooled neurons. We use Pearson's correlation to estimate the association degree between two neurons. We compute the correlation matrix among features and labels, and derive the degree of association of the pooled neurons from the degree of association between each pair of neurons in the previous layer. The pooling process is repeated over aggregated features and labels until a maximum number of pooling layers is reached.

Once the high-level features and labels are extracted from the dataset, they are connected together with one or several hidden processing layers. These hidden layers are equipped with either ReLU, sigmoid or hyperbolic tangent transfer functions, therefore conferring the neural system with prediction capabilities. Finally, a decoding process [5] is performed, which connects the high-level labels to the original ones by means of one or more hidden processing layers.

3 Computing the degree of association among neurons from granulation entropy

In [2], we present a new method that replaces the correlation measure (i.e. that quantifies the association between two neurons) with another one that computes the entropy in the information granules that are generated from two features or labels. Unlike the pooling approach proposed in [1], this proposal does not require that either the features or labels have a certain degree of correlation with each other. The rationale behind the proposal suggests that two features (or labels) can be associated if the granulations generated from them have equal entropy [9]. Therefore, the proposal consists in obtaining a universe granulation, where each feature (or label) defines an indiscernibility relation, and the information granules are the set of indiscernible objects with respect to the feature (or label) under consideration. In this way, it is verified if the coverings (or partitions) generated by two features (or labels) induce similar entropy values.

4 Concluding Remarks

The numerical simulations on several MLC datasets show a significant reduction in the number of problem features and labels (i.e. a reduction of up to 96% and 87%, respectively), without affecting network's discriminatory capability. Having a smaller neural system implies that the training time is smaller when compared with a model that uses the full set of features and labels. Despite of the relatively good results reported by the model in [1], the function used to quantify the association between problem variables does not seem to be suitable for datasets having poor correlation among their features or labels. In order to mitigate this, a variant based on granulation entropy in [2] is proposed.

References

1. Bello, M., Nápoles, G., Sánchez, R., Bello, R., Vanhoof, K.: Deep neural network to extract high-level features and labels in multi-label classification problems. Neurocomputing **413**, 259–270 (2020)
2. Bello, M., Nápoles, G., Sánchez, R., Vanhoof, K., Bello, R.: Feature and label association based on granulation entropy for deep neural networks. In: International Joint Conference on Rough Sets. pp. 225–235. Springer (2020)
3. Charte, F., del Jesus, M.J., Rivera, A.J.: Multilabel classification: problem analysis, metrics and techniques. Springer (2016)
4. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)
5. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science **313**(5786), 504–507 (2006)
6. Lee, C.Y., Gallagher, P.W., Tu, Z.: Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In: Artificial intelligence and statistics. pp. 464–472 (2016)
7. Wan, S., Duan, Y., Zou, Q.: Hpslpred: an ensemble multi-label classifier for human protein subcellular location prediction with imbalanced source. Proteomics **17**(17-18), 1700262 (2017)
8. Xia, Y., Chen, K., Yang, Y.: Multi-label classification with weighted classifier selection and stacked ensemble. Information Sciences (2020)
9. Yao, Y.: Probabilistic approaches to rough sets. Expert systems **20**(5), 287–297 (2003)

GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series

Edward De Brouwer*, Jaak Simm*, Adam Arany, and Yves Moreau

ESAT-STADIUS
KU Leuven
Leuven, 3001, Belgium
edward.debrouwer,jaak.simm,adam.arany,yves.moreau@esat.kuleuven.be

1 Introduction

Multivariate time series are ubiquitous in various domains of science [2, 6, 7] and much of the methodology for time-series analysis assumes that signals are measured systematically at fixed time intervals. However, much real-world data can be *sporadic* (*i.e.*, the signals are sampled irregularly and not all signals are measured each time). A typical example is patient measurements, which are taken when the patient comes for a visit (*e.g.,* sometimes skipping an appointment) and where not every measurement is taken at every visit. Modeling then becomes challenging as such data violates the main assumptions underlying traditional machine learning methods (such as recurrent neural networks).

Recently, the Neural Ordinary Differential Equation (ODE) model [1] opened the way for a novel, continuous representation of neural networks. As time is intrinsically continuous, this framework is particularly attractive for time-series analysis. It opens the perspective of tackling the issue of irregular sampling in a natural fashion, by integrating the dynamics over whatever time interval needed. Up to now however, such ODE dynamics have been limited to the continuous *generation* of observations (*e.g.*, decoders in variational auto-encoders (VAEs) [4] or normalizing flows [5]).

Instead of the encoder-decoder architecture where the ODE part is decoupled from the input processing, we introduce a tight integration by *interleaving* the ODE and the input processing steps. Conceptually, this allows us to drive the dynamics of the ODE directly by the incoming sporadic inputs. To this end, we propose (1) a continuous time version of the Gated Recurrent Unit and (2) a Bayesian update network that processes the sporadic observations. We combine these two ideas to form the GRU-ODE-Bayes method.

The tight coupling between observation processing and ODE dynamics allows the proposed method to model fine-grained nonlinear dynamical interactions between the variables. As illustrated in Figure 1, GRU-ODE-Bayes can (1) quickly infer the unknown parameters of the underlying stochastic process and (2) learn the correlation between its variables (red arrows in Figure 1). In contrast, the encoder-decoder based method NeuralODE-VAE proposed by [1] captures the

* Equal contribution

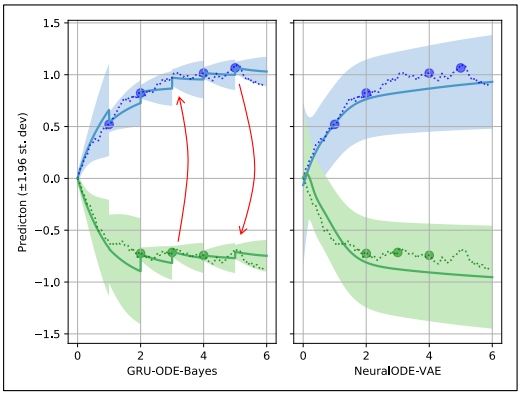


Fig. 1. Comparison of GRU-ODE-Bayes and NeuralODE-VAE on a 2D Ornstein-Uhlenbeck process with highly correlated Wiener processes ($\rho = 0.99$). Dots are the values of the actual underlying process (dotted lines) from which the sporadic observations are obtained. Solid lines and shaded areas are the inferred means and 95% confidence intervals. Note the smaller errors and smaller variance of GRU-ODE-Bayes vs. NeuralODE-VAE. Note also that GRU-ODE-Bayes can infer that a jump in one variable also implies a jump in the other unobserved one (red arrows). Similarly, it also learns the reduction of variance resulting from a new incoming observation.

general structure of the process without being able to recover detailed interactions between the variables.

Our model enjoys important theoretical properties. We frame our analysis in a general way by considering that observations follow the dynamics driven by a stochastic differential equation (SDE). In this paper, we show that GRU-ODE-Bayes can exactly represent the corresponding Fokker-Planck dynamics in the special case of the Ornstein-Uhlenbeck process, as well as in generalized versions of it.

We further perform an empirical evaluation and show that our method outperforms the state of the art on healthcare and climate data. In healthcare, we used electronic health records (EHR) from the MIMIC-III clinical database [3], which contains EHR for more than 60,000 critical care patients. We select a subset of 21,250 patients with sufficient observations and extract 96 different longitudinal real-valued measurements over a period of 48 hours after patient admission. We predicted the next 3 vitals measurements of intensive care patients after 36 hours observations. For the climate application, we used the publicly available United State Historical Climatology Network (USHCN) daily data set ushcn, which contains measurements of 5 climate variables (daily temperatures, precipitation, and snow) over 150 years for 1,218 meteorological stations scattered over the United States. To showcase the capability of our approach, we artificially downsampled the available data and predicted future measurements based on 3 years observations.

References

1. Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.: Neural ordinary differential equations. In: Advances in Neural Information Processing Systems, 2018 (2018)
2. Jensen, A.B., Moseley, P.L., Oprea, T.I., Ellesøe, S.G., Eriksson, R., Schmock, H., Jensen, P.B., Jensen, L.J., Brunak, S.: Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. Nature communications **5**, 4022 (2014)
3. Johnson, A.E., Pollard, T.J., Shen, L., Li-wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: Mimic-iii, a freely accessible critical care database. Scientific data **3**, 160035 (2016)
4. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
5. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning. pp. 1278–1286 (2014)
6. Scargle, J.D.: Studies in astronomical time series analysis. ii-statistical aspects of spectral analysis of unevenly spaced data. The Astrophysical Journal **263**, 835–853 (1982)
7. Schneider, T.: Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. Journal of climate **14**(5), 853–871 (2001)

On the State Space of Fuzzy Cognitive Maps using Shrinking Functions

Leonardo Concepción[1,2], Gonzalo Nápoles[2,3], Rafael Bello[1], and Koen Vanhoof[2]

[1] Department of Computer Science, Universidad Central de Las Villas, Cuba
[2] Faculty of Business Economics, Hasselt Universiteit, Belgium
[3] Department of Cognitive Science & Artificial Intelligence, Tilburg University, The Netherlands

1 Introduction

Fuzzy Cognitive Maps (FCMs) [4, 5] are recurrent neural networks for modeling complex systems. Existing theoretical studies on FCMs are mainly devoted to convergence issues, commonly covering the existence and uniqueness of fixed points [1, 3, 6]. Other results reported in [7–10] address the convergence of FCM models used in prediction/classification scenarios.

Concerning the theoretical analysis of FCMs' dynamics, we summarize our paper *Unveiling the Dynamic Behavior of Fuzzy Cognitive Maps* [2]. First, we introduce several definitions and theorems that allow studying the dynamic behavior of FCMs equipped with monotonically increasing functions bounded into non-negative intervals. The strong version of our theorem proves that the state space of an FCM shrinks infinitely and converges to a so-called *limit state space*, which could be a fixed-point attractor in some cases. This allows envisaging, to some extent, the FCM model's behavior before the inference stage. As a second contribution, we explore the covering and proximity of *feasible activation spaces*, which help explain why FCMs sometimes perform poorly when solving complex prediction problems. In other words, we show why we should not expect impressive prediction rates when the model has low covering values as the FCM feasible state space is small.

2 Shrink Functions and State Space Estimation in FCM-based Models

We define F as the set of all monotonically increasing functions bounded into non-negative intervals. Also, let $f_i \in F$ be the transfer function used in the activation process of neuron C_i in the FCM. In [2], we refer to an F-function as any function belonging to F.

Let $\mathcal{H}_W$ and $\mathcal{H}_T$ be functions that take an FCM-based model $\mathcal{M}$ and a feasible state space at the t-th iteration $\mathcal{S}^{(t)}$ for this map and return a feasible state space at the $(t+1)$-th iteration $\mathcal{S}^{(t+1)}$ for the same map. While $\mathcal{H}_W$ uses the

weight matrix W of $\mathcal{M}$ to calculate a feasible state space for the $(t+1)$-th iteration, $\mathcal{H}_T$ uses the FCM's topology only. Based upon estimated bounds for the successive activation values and from the monotonically increasing property of $f_i \in F$, we assert that over the same FCM, these two shrink functions transform feasible state spaces into state spaces which are also feasible.

To show that FCMs are not completely unpredictable, we propose two theorems as the pillars of our state-space estimation: the *Weak Shrinking State Space (WSSS)* and the *Strong Shrinking State Space (SSSS)*. The former asserts that the state spaces shrink from one iteration to the next one, although it is possible that $\mathcal{S}^{(t)} = \mathcal{S}^{(t+1)}$, which would imply that $\mathcal{S}^{(t)} = \mathcal{S}^{(t+k)}\ \forall k \in \mathbb{N}$. So, the state spaces may not shrink forever. The latter only varies in the sense that transfer functions are now bounded into open intervals. This means that the state space bounds are never reachable and hence, the state spaces will shrink forever and they will have a limit. The *limit state space* of $\mathcal{M}$ is $\mathcal{S}^{(\infty)} = \lim_{t\to\infty} \mathcal{S}^{(t)}$, when state spaces are iteratively calculated using either shrink function $\mathcal{H}_T$ or $\mathcal{H}_W$. According to simulations, $\mathcal{S}^{(\infty)}$ often contains a single point.

3 Covering and Proximity of FCM Models

In this section, we discuss two evaluation measures that help understand the properties of FCM-based systems. The *covering* quantifies the proportion of the induced activation space that is reachable by the neuron's activation values and the *proximity* measures the mean relative distance of neuron's activation values to the feasible activation spaces.

The results confirmed that better predictions for FCMs' behavior arise working with stable maps and their weight sets. Small covering values are evidence of the reduced representativeness of induced activation space, but sometimes we desire high covering values to represent the most diverse sets of outputs. As illustrated, such measures have a straightforward connection with the *SSSS Theorem*. More importantly, they help explain why FCMs sometimes perform poorly when applied to prediction problems that demand high accuracy.

4 Concluding Remarks

In [2], we have introduced a theoretical formalism consisting of definitions and theorems to unveil the dynamical behavior of FCMs equipped with transfer F-functions, from the perspective of their state spaces.

The *SSSS Theorem* enunciated in this paper ensures that the feasible state space of the targeted FCMs shrinks infinitely, yet the system converges to its limit state space. As shown in the experiments, approximating an FCM's limit state space is useful to predict fixed-point attractors. Likewise, we illustrated that the covering of feasible activation spaces is often poor and irregular for FCMs with reduced network topologies. This knowledge could be injected into the learning procedure in order to improve network's performance.

References

1. Boutalis, Y., Kottas, T.L., Christodoulou, M.: Adaptive estimation of fuzzy cognitive maps with proven stability and parameter convergence. IEEE Transactions Fuzzy Systems **17**(4), 874–889 (2009)
2. Concepción, L., Nápoles, G., Falcon, R., Vanhoof, K., Bello, R.: Unveiling the dynamic behavior of fuzzy cognitive maps. IEEE Transactions on Fuzzy Systems (2020)
3. Harmati, I.Á., Hatwágner, M.F., Kóczy, L.T.: On the existence and uniqueness of fixed points of fuzzy cognitive maps. In: Medina, J., Ojeda-Aciego, M., Verdegay, J.L., Pelta, D.A., Cabrera, I.P., Bouchon-Meunier, B., Yager, R.R. (eds.) Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations. pp. 490–500. Springer International Publishing (2018)
4. Kosko, B.: Fuzzy cognitive maps. International Journal Man-Machine Studies **24**(1), 65–75 (1986)
5. Kosko, B.: Hidden patterns in combined and adaptive knowledge networks. International Journal of Approximate Reasoning **2**(4), 377–393 (1988)
6. Kottas, T., Boutalis, Y., Christodoulou, M.: Bi-linear adaptive estimation of fuzzy cognitive networks. Applied Soft Computing **12**(12), 3736–3756 (2012)
7. Nápoles, G., Bello, R., Vanhoof, K.: How to improve the convergence on sigmoid fuzzy cognitive maps? Intelligent Data Analysis **18**(6S), S77–S88 (2014)
8. Nápoles, G., Concepción, L., Falcon, R., Bello, R., Vanhoof, K.: On the accuracy-convergence tradeoff in sigmoid fuzzy cognitive maps. IEEE Transactions on Fuzzy Systems **26**(4), 2479–2484 (2018)
9. Nápoles, G., Papageorgiou, E., Bello, R., Vanhoof, K.: On the convergence of sigmoid fuzzy cognitive maps. Information Sciences **349-350**, 154–171 (2016)
10. Nápoles, G., Papageorgiou, E., Bello, R., Vanhoof, K.: Learning and convergence of fuzzy cognitive maps used in pattern recognition. Neural Processing Letters **45**, 431–444 (2017)

Alternating Maximization with Behavioral Cloning

Aleksander Czechowski[1 [0000−0002−6054−9842]] and Frans A. Oliehoek[1 [0000−0003−4372−5055]]

Department of Intelligent Systems, Delft University of Technology, Delft, The Netherlands
{a.t.czechowski,f.a.oliehoek}@tudelft.nl

1 Introduction

The key difficulty of cooperative, decentralized planning lies in making accurate predictions about the behavior of one's teammates. In this paper we introduce a planning method of *Alternating maximization with Behavioural Cloning* (ABC) – a trainable online decentralized planning algorithm based on Monte Carlo Tree Search (MCTS), combined with models of teammates learned from previous episodic runs. Our algorithm relies on the idea of alternating maximization, where agents adapt their models one at a time in round-robin manner. Under the assumption of perfect policy cloning, and with a sufficient amount of Monte Carlo samples, successive iterations of our method are guaranteed to improve joint policies, and eventually converge.

2 The ABC method

Our planning algorithm is suitable for fully observable cooperative environments known as Multi-agent Markov Decision Processes (MMDPs). The setting is fully cooperative, and each agent is assumed to receive the same reward at each execution step of an episodic run. The planning is performed in a decentralized manner, and without communication between the agents. Each agent is equipped with an instance of the MCTS algorithm, a set of models of policies of its teammates, and a simulator of the environment. At each episodic step, each agent samples the simulator and teammate models to construct the tree of possible futures, estimate expected episodic rewards for individual actions, and choose the one which appears most beneficial.

Initially, agents are equipped with heuristic models of their teammates. They are assumed to act in a given environment repeatedly, for some large amount of episodic runs – either from simulation, or actual execution. Then, the agents use these experiences to learn to predict the actions of their colleagues. More specifically, every N episodic runs are grouped into one generation, and after each generation, the state-action episodic data, is used to train new agent models represented by convolutional neural networks; these are in turn provided to one of the agents, as the updated teammate models. At each generation only one agent updates its teammate models, which stabilizes training and, under certain assumptions on policy cloning, causes rewards to increase monotonically across the generations.

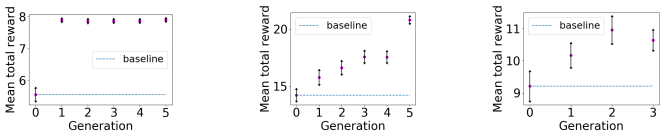


Fig. 1: Results from the factory floor experiment, in order of increasing difficulty. Left: two robots and preallocated tasks, middle: four robots and preallocated tasks, right: four robots and randomly appearing tasks. The baseline is the decentralized MCTS planning algorithm, introduced in [3].

3 Experiments

We test the efficiency of the algorithm by performing experiments in the spatial task allocation environment introduced in [2]. The domain consists of a gridworld-like planar map, where each position can be occupied by (cleaning) robots and tasks (e.g. litter). Each robot can perform either a movement action, which shifts the position of the robot accordingly, or a cleaning action, which removes one task at the current position. Attempted actions may succeed or not, according to predefined probabilities. Experiments show the effectiveness of the method, as an improvement across generations is observed, see Figure 1.

Acknowledgments

This project received funding from EPSRC First Grant EP/R001227/1, and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

References

1. Aleksander Czechowski and Frans A. Oliehoek. Decentralized MCTS via Learned Teammate Models Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pages 81–88, 2020.
2. Daniel Claes, Philipp Robbel, Frans A. Oliehoek, Karl Tuyls, Daniel Hennes, and Wiebe Van der Hoek. Effective approximations for multi-robot coordination in spatially distributed tasks. Proceedings of the Fourteenth International Conference on Autonomous Agents and Multiagent Systems, pages 881–890, 2015.
3. Daniel Claes, Frans A. Oliehoek, Hendrik Baier, and Karl Tuyls. Decentralised online planning for multi-robot warehouse commissioning. Proceedings of the Sixteenth International Conference on Autonomous Agents and Multiagent Systems, pages 492–500, 2017.

Abstract: Layered Integration Approach for Multi-view Analysis of Temporal Data* **

Michiel Dhont[1,2][0000−0002−5679−0991], Elena Tsiporkova[1], and Veselka Boeva[3][0000−0003−3128−191X]

[1] Sirris, Bd A. Reyerslaan 80, 1030 Brussels, Belgium
[2] Department of Electronics and Information Processing (ETRO), VUB, Belgium
[3] Blekinge Institute of Technology, Sweden
{michiel.dhont,elena.tsiporkova}@sirris.be, veselka.boeva@bth.se

Keywords: Data Integration · Data Mining · Temporal Data Clustering · Multi-view Learning.

1 Introduction

Mining data collected from continuous monitoring of industrial assets in the field allows to derive relevant insights about their operations and performance. Such complex real-world datasets are usually composed of heterogeneous subsets (or multi-views) of parameters, which should be considered explicitly during analysis in order to exploit fully the richness of the data. For instance, the performance of an industrial asset is impacted by a diverse set of factors e.g. operating modes concerned with the internal working of the asset and exogeneous factors such as weather conditions. However, it is not trivial to directly link or trace back certain performance to distinct operating modes due to the multitude of influencing factors, which are often also highly interdependent.

In addition, real-world datasets often originate from different sources, which may differ in period coverage, resolution, data quality, technical configuration, etc. Pooling multi-source datasets together, which is often done to increase statistical representativeness, requires standardization and normalization, which often leads to information loss and may mask source-specific features. For instance, mining for distinct operating modes is more appropriate to be pursued per asset, rather than pooling everything together, since not all assets might go through all operating modes. This implies that one might need to approach multi-source analysis in an incremental fashion rather than aiming for brute force integration of all the available data.

2 Proposed Approach

Classical data mining and analysis approaches have still some shortcomings in this aspect aiming at delivering a total integration solution at once. An alter-

* This research was supported by the Brussels-Capital Region - Innoviris, and received funding from the Flemish Government (AI Research Program) and BitWind project.
** The full paper is accepted for the 5^{th} AALTD workshop and publication in LNAI.

native approach is to **exploit the multi-view nature of the data**. Some rewarding techniques of multi-view mining have been already proposed in the literature [1, 2]. However, they all were concerned with single-source datasets and dedicated to one specific mining approach (e.g. clustering or deep learning). This research provides a general analysis methodology, which focuses on the following key aspects: initial *individual analysis* per source in order to preserve the richness and the authenticity of each source; individual *mediation analysis* per source aiming at bringing the sources closer together; cross-source *integration analysis* aiming at leveraging analysis results across the sources without compromising their individual characteristics.

More concretely, the proposed approach consists of several distinctive layers: (i) select a suitable set (view) of parameters in order to identify characteristic behaviour within each individual source (ii) exploit an alternative set (view) of raw parameters (or high-level features) to derive some complementary representations (e.g. related to source performance) of the results obtained in the first layer with the aim to facilitate comparison and mediation across the different sources (iii) integrate those representations in an appropriate way, allowing to trace back similar cross-source performance to certain characteristic behaviour of the individual sources.

3 Implementation and Results

The validity and the potential of the proposed approach have been demonstrated on a real-world dataset of a fleet of wind turbines. We have been able to identify distinctive profiles of production performance and subsequently, have been able to establish an explicit link between those performance profiles and well characterised operating modes. Subsequently, distinctive performance profiles have been derived and associated with each operating mode, which enable converting the fleet data into powerful letter code suitable for more advanced mining.

4 Conclusion

We have proposed a novel data analysis approach that can be used for multi-view analysis and integration of heterogeneous real-world datasets originating from multiple sources. The validity and the potential of the proposed approach has been demonstrated on a real-world dataset of a fleet of wind turbines. The obtained results are very encouraging. The method is very efficient and robust in detecting characteristic operating modes across the fleet.

References

1. Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.
2. Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

Mining Constrained Regions of Interest: An Optimization Approach*

Alexandre Dubray, Guillaume Derval, Siegfried Nijssen, and Pierre Schaus

UCLouvain - ICTEAM/INGI, Louvain-la-Neuve, Belgium
{first}.{second}@uclouvain.be

The number of devices producing spatiotemporal data is increasing and so the data that need to be processed by the applications. It is important to have generic and flexible tools to be able to summarize and analyze this data efficiently. One such tool is the discovery of Regions of Interest (ROI) or densely visited regions. The discovery of ROI can also be instrumental as a preprocessing step to rewrite the trajectories as a sequence of interesting places instead of a sequence of GPS data point, which are easier to process and match by frequent sequence analysis tools [1].

There exists different algorithms to extract ROIs, depending on the definition of density we choose. In our paper we are interested in grid-based approaches which first divide the map into a grid and assign a density value to each cell. The density of a cell is the number of trajectories crossing it. A cell is dense when its density value is above a threshold. Then the ROIs are formed by aggregating, in some ways, the dense cells, similarly to a clustering algorithm. Multiple approaches exist to aggregate the dense cells in ROIs, generally based on a greedy expansion or clustering. While these methods give good results, they do not easily accept new constraints such as various types of shapes, and application dependent intra- or inter-ROI constraints.

To address these weaknesses, we propose a new approach to extract *Constrained Regions of Interest*, which is illustrated in Figure 1. This two-step optimization process allows to impose constraints on individual ROI (*intra-ROI* constraints) and between the ROIs (*inter-ROI* constraints).

The first step is to generate a set of candidates ROI (e.g. rectangles, circles, or any non-parametric shapes) that respects the *intra-ROI* constraints such as "Every ROI must contain at least one cultural point of interest".

Once we have this set of candidates, we will select the ROIs to return to the user, while ensuring they respect *inter ROIs* constraints. The idea is to consider that a set of ROI is a classifier that indicates whether a cell on the grid is dense or not. Such classifier would make some errors (a dense cell not covered by a ROI or a non-dense cell covered by a ROI) and we are interested in a set of K ROIs that minimizes the number of errors.

We propose an efficient *Integer Linear Program* (ILP) model to solve this problem with one binary decision variable per candidate with the constraint that two ROIs cannot overlap. We use the *Minimum Description Length* principle to automatically detect the appropriate K. During this phase we can add

* This paper is a summary of a paper with the same name published at Discovery Science 2020 https://doi.org/10.1007/978-3-030-61527-7_41.

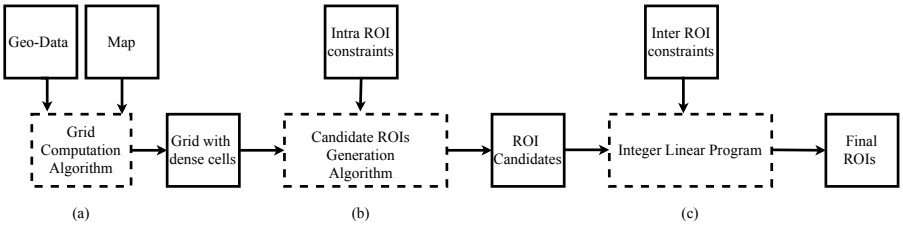


Fig. 1: Process of our approach: a) Creation of the density grid from the trajectories; b) Generation of candidates shape on the grid; c) Selection of the ROIs from the set of candidates.

constraints between the ROI as long as they can be translated into a linear constraint. For example, the constraint "If we select this ROI, then that ROI must also be selected" can be expressed using classical boolean logic constraints on the decision variables.

We compared our method to *PopularRegions* [2], a method specifically designed to extract ROIs, and OPTICS [3] when clustering the dense cells. We show that our method is slower than the others but, as the number of candidates decreases, our run time becomes like OPTICS. Moreover we obtain a lower description length with a better balance between the errors and the number of ROIs. This allows our method to be more flexible and have a better generalization. Finally, we show that when adding up to 40% of noise in the data, our method is more stable than *PopularRegion* and OPTICS. We refer to the complete article [4] for further information on the approach.

References

1. John OR Aoga, Tias Guns, and Pierre Schaus. Mining time-constrained sequential patterns with constraint programming. *Constraints*, 22(4):548–570, 2017.
2. Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *SIGKDD*, 2007.
3. Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *ACM Sigmod record*, 1999.
4. Alexandre Dubray, Guillaume Derval, Siegfried Nijssen, and Pierre Schaus. Mining constrained regions of interest: An optimization approach. In *International Conference on Discovery Science*, pages 630–644. Springer, 2020.

Extended Abstract: An Interpretable Semi-supervised Classifier using Rough Sets for Amended Self-labeling*

Isel Grau[1], Dipankar Sengupta[1,2], Maria M. Garcia Lorenzo[3], and Ann Nowe[1]

[1] Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium
[2] Centre for Cancer Research and Cell Biology, Queen's University Belfast, UK
[3] Department of Computer Science, Universidad Central de Las Villas, Cuba

This document is an extended abstract of the paper accepted at the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), held from the 19th to the 24th of July, 2020, Glasgow, United Kingdom.

Gathering data examples for training a machine learning classifier in a real-world scenario is often simple. However, the process of assigning labels to the examples can be costly in terms of money, time, or effort. In such scenarios, we might obtain datasets with more unlabeled than labeled data. Semi-supervised classification [4] techniques arise from the need to address this problem using both labeled and unlabeled data for training a classifier. The aim is to increase the classifier's generalization ability compared to a supervised classifier that only uses the available labeled data.

On the other hand, an increasing requirement observed in machine learning is to obtain not only precise models but also interpretable ones. End users often demand an insight into how an algorithm arrives at a particular outcome and needs an explanation of the decisions. A certain degree of global interpretability can be obtained using more transparent techniques as proxies for solving a task [1]. We refer to intrinsically interpretable models (e.g., linear regression, decision trees or decision lists) as white boxes, as opposed to the less interpretable black-box ones. Grey-box models use white boxes as surrogates for distilling previously trained black boxes. The grey boxes attempt to explain the domain by approximating the predictions produced by a black-box classifier, in an intrinsically interpretable structure.

In this paper, we explore the performance of the *self-labeling grey-box* (SlGb) [2]. In the SlGb, we use a black-box classifier to predict the decision class of the unlabeled instances, while a surrogate white box is used to build an interpretable predictive model, based on the whole instance set. The aim is to outperform the base white-box component using only the available labeled data,

* This work was supported by the IMAGica project, financed by the Interdisciplinary Research Programs and Platforms (IRP) funds of the Vrije Universiteit Brussel; and the BRIGHTanalysis project, funded by the European Regional Development Fund (ERDF) and the Brussels-Capital Region as part of the 2014-2020 operational program through the F11-08 project ICITY-RDI.BRU (icity.brussels).

while maintaining a good balance between performance and interpretability. The SlGb approach's performance largely depends on the black-box classifier's prediction capability when classifying unseen instances. In the context of self-labeling, the classification mistakes can reinforce themselves if no amending procedure is used during self-training. Therefore, we explore the effect of two amending procedures for assigning more importance to more reliable instances before training the surrogate white box, avoiding the propagation of errors or inconsistent information. The first strategy is based on class membership probabilities provided by the black box in the self-labeling. The second strategy aims to correct the inconsistency in the labels in the enlarged dataset by computing the certainty of the classification based on the Rough Set Theory (RST) [3] inclusion degree measure.

The experiments show that the choice of a white box and amending is relevant for the size of the structure. SlGb produces simpler models when using decision lists instead of a C4.5 decision tree as surrogate white boxes, even when no amending is performed. However, the amending procedures help further increase the simplicity without affecting the prediction rates by giving more importance to confident instances in the self-labeling. Especially RST based amending looks more promising since it does not need the black-box base classifier to provide calibrated probabilities. Furthermore, RST-based amending could be the right choice for a given case study where the uncertainty coming from inconsistency is high, even on the available labeled data. The study varying the number of unlabeled instances and labeled instances together shows that even when the number of labeled instances is not that scarce, the SlGb is able to leverage unlabeled instances for increasing the performance. Another conclusion is that adding unlabeled instances does not make the interpretability worse compared to adding more labeled instances. This evidences that the RST-based amending avoids that the SlGb generates more rules from inconsistent instances. Finally, the experimental comparison shows that our SlGb method outperforms the state-of-the-art self-labeling approaches, yet being far more simple in structure than these techniques.

References

1. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)
2. Grau, I., Sengupta, D., Lorenzo, M.M.G., Nowe, A.: Interpretable self-labeling semi-supervised classifier. In: Proceedings of the IJCAI/ECAI 2018 2nd Workshop on Explainable Artificial Intelligence. pp. 52–57 (2018)
3. Pawlak, Z.: Rough sets. International Journal of Computer & Information Sciences **11**(5), 341–356 (1982)
4. Zhu, X., Goldberg, A.: Introduction to semi-supervised learning. Morgan & Claypool Publishers (2009)

Reinforcement learning for personalization: a systematic literature review (Abstract)

Floris den Hengst[1], Eoin Martino Grua[⋆,1], Ali el Hassouni[⋆,1], and Mark Hoogendoorn[⋆,1]

Vrije Universiteit Amsterdam, <initial>.<affix>.<surname>@vu.nl

Abstract. This compressed contribution presents a survey into reinforcement learning (RL) for personalization.

Keywords: Reinforcement Learning, Contextual Bandits · Personalization.

1 Introduction

When products and services are adapted to individual tastes, they become more appealing, desirable, informative, etc. to the intended user than one-size-fits all alternatives. Digital systems enable such *personalization* on a grand scale. The key enabler is data. While the software is identical for all users, the system's behavior can be tailored based on experiences with individual users. Reinforcement learning (RL) has been attracting attention for personalization. An overview of RL for personalization, however, is lacking.

This contribution summarizes our systematic review and categorization of 166 papers describing RL solutions to personalization problems, problem contexts and evaluation strategies across domains [1]. It thus aids researchers and practitioners in identifying relevant related work, promotes the understanding of the usage of RL and identifies challenges across domains. The data used and a tool[1] for exploring it have been made available.

2 Systematic Literature Review

We performed a systematic literature review following the PRISMA standards. We queried five databases on keywords similar to 'reinforcement learning', 'contextual bandit' and 'personalization' and found 983 publications. Titles and abstracts and subsequently full texts were assessed for eligibility, resulting in 166 included papers. For included papers, data on the problem context, solution architecture and evaluation strategy were extracted.

⋆ Authors contributed equally

[1] Data exploration tool at https://florisdh.nl/rl4personalization/

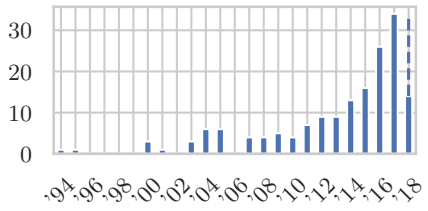


Fig. 1: Number of publications per year, dashed indicates projection for full year.

Domain	#	Domain	#
Health	44	Transport	9
Entertainment	30	Energy	6
Commerce	28	Other	5
Education	25	Smart Home	4
Domain Indep.	11	Communication	4

Table 1: Number of publications per domain.

3 Results

Figure 1 shows a marked increase in publications over time. Table 1 shows that RL is used for personalization in various domains. We continue by looking at problem contexts, solution patterns and evaluation strategies.

Problem In most publications (130/166), users do not provide feedback to the system explicitly, but feedback is derived from various measurements that indicate suitability of system behavior. Data on user responses are available in a minority of cases (66/166) and safety concerns are mentioned in a reasonable number of works (30/166) whereas privacy is not (9/166).

Solutions The RL framework can be used for personalization in various ways: learning a single policy across all users is the most popular approach (91/166). User traits can be included in the state representation in this approach (51/91). The next most popular approach is to represent each user as a separate environment and learn a policy per user (59/166). Hybrid approaches, such as a policy for every group of users, are less popular (11/166). Only a small fraction compares different approaches (5/166). Combining these approaches is an interesting direction for future work, e.g. to increase the level of personalization as more data is obtained

When analyzing the most popular algorithms, we found generic and well-established ones such as Q-learning, contextual bandits and Sarsa to be most popular. Besides these, we find little algorithm re-use. In recent years, approaches that include function approximation with deep neural networks such as DQN or DDQN are becoming more popular.

Evaluation We reviewed the usage of live or real-life data for evaluation and found that the number of studies with such an evaluation is increasing. This indicates that RL has become sufficiently robust to apply in contexts that involve humans. However, the relative number of works that include a realistic evaluation is not increasing. Furthermore, we find that little works compare multiple algorithms. This indicates that the field is growing, but not yet maturing.

References

1. den Hengst, F., Grua, E.M., el Hassouni, A., Hoogendoorn, M.: Reinforcement learning for personalization: A systematic literature review. Data Science (Preprint), 1–41

Towards Partial Order Reductions for Strategic Ability (abstract)

Wojciech Jamroga[1,2], Wojciech Penczek[1], Teofil Sidoruk[1,3], Piotr Dembiński[1], and Antoni Mazurkiewicz[1]

[1] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
[2] Interdisciplinary Centre for Security, Reliability, and Trust, SnT, University of Luxembourg, Luxembourg
[3] Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland

Strategic Ability, Asynchronous Agent Systems, and Partial-Order Reductions

Alternating-time temporal logic **ATL*** and its fragment **ATL** [1] extend temporal logic with the notion of *strategic ability*. They allow to express statements about what agents (or groups of agents) can achieve. For example, $\langle\langle i \rangle\rangle \mathrm{F}\, \mathsf{win}_i$ says that agent i can eventually win no matter what the other agents do. Such properties can be useful for specification, verification, and reasoning about interaction in agent systems.

In this paper, we make the first step towards strategic analysis of *asynchronous* multi-agent systems. Our contribution is threefold. First, we define a semantics of strategic abilities for agents in asynchronous systems, with and without perfect information. Secondly, we present some general complexity results for verification of strategic abilities in such systems. Thirdly, and most importantly, we adapt *partial order reduction (POR)* to model checking of strategic abilities for agents with imperfect information. We also present experimental results demonstrating that POR allows to significantly reduce the size of the model, and thus to make the verification more feasible. In fact, we show that the most efficient variant of POR, defined for linear time logic **LTL**, can be applied almost directly. The (nontrivial) proof that the **LTL** reductions work also for the more expressive strategic operators is the main contribution of this paper. Interestingly, the scheme does *not* work for verification of agents with perfect information.

Conclusions in a Nutshell

The theoretical complexity results follow the same pattern as those for synchronous MAS, though proving them required careful treatment. Consequently, model checking of strategic abilities under imperfect information for asynchronous systems is as hard as in the synchronous case. This makes model reductions essential for practical verification. The most important result of this paper consists

in showing that the partial order reduction for $\mathbf{LTL}_{-\mathbf{X}}$ can be almost directly applied to $\mathbf{ATL}_{\mathrm{ir}}$ without nested strategic modalities. The importance of the result stems from the fact that $\mathbf{LTL}_{-\mathbf{X}}$ has relatively weak distinguishing power, and therefore admits strong reductions, clustering paths into relatively few equivalence classes.

Interestingly, it turns out that the scheme does *not* work for $\mathbf{ATL}^*$ with perfect information strategies. Until now, virtually all the results have suggested that verification of strategic abilities is significantly easier for agents with perfect information. Thus, we identify an aspect of verification that might be in favour of imperfect information strategies in some contexts.

All the technical details can be found in the original paper [4].

Acknowledgements

In memory of Prof. Piotr Dembiński (1940-2020). W. Jamroga and W. Penczek acknowledge the support of the National Centre for Research and Development, Poland (NCBR), and the Luxembourg National Research Fund (FNR), under the PolLux/FNR-CORE projects VoteVerif (POLLUX-IV/1/2016) and STV (POLLUX-VII/1/2019). W. Penczek and T. Sidoruk acknowledge also the support of the French National Centre for Scientific Research (CNRS), and the Polish Academy of Sciences (PAN), under the CNRS/PAN project PARTIES.

References

1. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.
2. N. Bulling, J. Dix, and W. Jamroga. Model checking logics of strategic ability: Complexity. In M. Dastani, K. Hindriks, and J.-J. Meyer, editors, *Specification and Verification of Multi-Agent Systems*, pages 125–159. Springer, 2010.
3. P. Godefroid. Using partial orders to improve automatic verification methods. In *Proceedings of the 2nd International Conference on Computer Aided Verification, CAV '90, New Brunswick, NJ, USA, June 18-21, 1990*, pages 321–340, 1991.
4. W. Jamroga, W. Penczek, T. Sidoruk, P. Dembiński, and A. Mazurkiewicz. Towards partial order reductions for strategic ability. *Journal of Artificial Intelligence Research*, 68:817–850, 2020.
5. D. Peled. All from one, one for all: On model checking using representatives. In *Proceedings of the 5th International Conference on Computer Aided Verification, CAV '93, Elounda, Greece, June 28 - July 1, 1993*, pages 409–423. Springer-Verlag, 1993.
6. D. Peled. Combining partial order reductions with on-the-fly model-checking. *Formal Methods in System Design*, 8(1):39–64, 1996.
7. P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.

An ideal team is more than a team of ideal agents*

Can Kurtan, Pınar Yolum, and Mehdi Dastani

Utrecht University, The Netherlands
{a.c.kurtan, p.yolum, m.m.dastani}@uu.nl

The problem of forming teams has been a central question in many disciplines. Put simply, the problem is to form a set of agents with required capabilities so that they can perform a task together. In the domain of service composition, this would correspond to a set of service providers, each performing a single service, which overall yields a composition of the desired service. In the domain of query answering, this would correspond to data sources that can each answer part of a given question. In the domain of rescue operations, this would correspond to a set of human and robot agents that work to help civilians.

The term team may refer to a set of agents only or a set of agents and their specific subtasks assignments. Following this, here we refer to teams as the set of agents and their assigned subtasks. Assigning subtasks to individual agents is typically modeled as an optimization problem, where coverage of the subtasks is maximized, or the number of agents involved is minimized. Earlier approaches model team formation with three important assumptions: (i) the overall task can be divided into independent subtasks, (ii) agents' capabilities as to what subtask they can do is known or computed easily and (iii) the agents' capabilities of performing a subtask is binary (e.g., no variation in the quality of the subtask performance). Even under these assumptions, the problem of assigning subtasks to individual agents is known to be NP-hard.

In many domains, these assumptions do not hold. Consider the following simple example from a query answering domain, where the question is to find the title, author and summary of books. Three agents separately provide book names, book authors, and book summaries. An answer to the question can be found by using the information of these agents if they provide title, author and summary information of the same books. The individual performance of an agent depends on which subquery the agent is assigned. For example, one agent provides three book titles and a thousand book summaries, while another agent provides a thousand book titles and three book summaries. The number of books returned in the answer depends on which agents are assigned to which subqueries. Hence, building a team based on finding best performing individuals does not always yield the best performing team.

This paper summarizes our work [1] that addresses the above challenges: building teams for tasks whose subtasks are dependent to each other and the performance of agents is affected by the assignments. We represent the interplay

* This work was done in the context of the Golden Agents project, funded by the Netherlands Organization of Science NWO – Large Investments program.

between possible assignments of the agents using *expertise graph*, which stores information on how well agents perform tasks individually and how well they can support each other on common or different subtasks. It represents agent-capability pairs as nodes and pair-wise coperformances as edges. An edge between two nodes can be one of the three types: c-edge denoting how well the agent can carry out two subtasks, j-edge denoting two agents working on two different tasks, and s-edge denoting how well one agent supports the other on the same subtask.

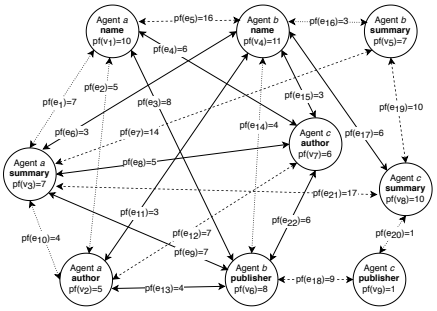


Fig. 1. An example expertise graph

We develop two algorithms: *one shot team building* and *iterative team building* algorithms. Our algorithms use the expertise graph to approximate how likely the agents are to perform well in dependent subtasks. We provide graph metrics, such as cooperativeness and versatility, to use as heuristics. The one shot team building algorithm selects assignments that have the highest value for a given metric. It is similar to the traditional algorithms that maximize one specific property of a team, such as communication. The iterative team building algorithm starts with an existing team for a task, which might be generated with a tool in hand, and improves the team to yield a better team performance in an iterative manner. It replaces an assigned agent with another agent that has better local cooperativeness in that specific team. The algorithm uses performance estimation functions to decide when to iterate to find a better team or to stop.

We demonstrate the workings of our algorithms in a query answering multiagent system, where agents are data providers and tasks are queries. We evaluate the algorithms in an experimental setup and compare the performance of the algorithms. We see that for smaller task sizes, the teams built by the one shot algorithm obtain better results. However, for larger task sizes, the iterative algorithm outperforms the one shot algorithm consistently. This shows that when the task is large, building a team by just adding "ideal" agents does not capture the relations among them. It is necessary to consider how the team will perform as a whole and update the team when an agent does not fit the team. Our work differs from other approaches in that we consider how well two agents work together on a given subtask as well as the possible degradation of performance when multiple subtasks are performed by a single agent. Further, we can build better performing teams by improving existing teams incrementally.

References

1. Kurtan, C., Yolum, P., Dastani, M.: An ideal team is more than a team of ideal agents. In: Proceedings of the 24th European Conference on Artificial Intelligence (ECAI). pp. 43–50 (2020)

Deep reinforcement learning for large-scale epidemic control

Pieter J.K. Libin[1,2,3], Arno Moonens[1], Timothy Verstraeten[1], Fabian Perez-Sanjines[1], Niel Hens[3], Philippe Lemey[2], and Ann Nowé[1]

[1] Vrije Universiteit Brussel, Brussels, Belgium
{pieter.libin,arno.moonens,timothy.verstraeten,fperezsa,ann.nowe}@vub.be
[2] KU Leuven, Leuven, Belgium
philippe.lemey@kuleuven.be
[3] Hasselt University, Hasselt, Belgium
{pieter.libin,niel.hens}@uhasselt.be

This paper[4] *was accepted and presented at the ECML-PKDD 2020 conference.*

Epidemics of infectious diseases are an important threat to public health and global economies. The most efficient way to combat epidemics is through prevention. To develop prevention strategies and to implement them as efficiently as possible, a good understanding of the complex dynamics that underlie these epidemics is essential. To properly understand these dynamics, and to study emergency scenarios, epidemiological models are necessary. Such models enable us to make predictions and to study the effect of prevention strategies in simulation. The development of prevention strategies, which need to fulfil distinct criteria (i.a., prevalence, mortality, morbidity, cost), remains a challenging process. For this reason, we investigate a deep reinforcement learning (RL) approach to automatically learn prevention strategies in an epidemiological model. The use of model-free deep reinforcement learning is particularly interesting, as it allows us to set up a learning environment in a complex epidemiological setting (i.e., large state space and non-linear dependencies) while imposing few assumptions on the policies to be learned. In this work, we conduct our experiments in the context of pandemic influenza, where we aim to learn optimal school closure policies to mitigate the epidemic [1].

Pandemic preparedness is important, as influenza pandemics have made many victims in the (recent) past and the ongoing COVID-19 epidemic is yet another reminder of this fact. Contrary to seasonal influenza epidemics, an influenza pandemic is caused by a newly emerging virus strain that can become pandemic by spreading rapidly among naive human hosts (i.e., human hosts with no prior immunity) worldwide. This means that at the start of the pandemic no vaccine will be available and it will take several months before vaccine production can commence. For this reason, learning optimal strategies of non-therapeutic intervention measures, such as school closure policies, is of great importance to mitigate pandemics. To meet this objective, we consider a reinforcement learning approach. However, as the state-of-the-art of reinforcement learning techniques requires many interactions with the environment in order to converge, our first

[4] https://bitbucket.org/ghentdatascience/ecmlpkdd20-papers/raw/master/ADS/sub_616.pdf

contribution entails a realistic epidemiological model that still has a favourable computational performance.

Specifically, we construct a meta-population model that consists of a set of 379 interconnected patches, where each patch corresponds to an administrative region in Great Britain and is internally represented by an age-structured stochastic compartmental model. To conduct our experiments, we establish a Markov decision process with a state space that directly corresponds to our epidemiological model, an action space that allows us to open and close schools on a weekly basis, a transition function that follows the epidemiological model's dynamics, and a reward function that is targeted to the objective of reducing the attack rate (i.e., the proportion of the population that was infected). In this work, we will use "Proximal Policy Optimization" (PPO) to learn the school closure policies.

First, we set up an experiment in an epidemiological model that covers a single administrative district. This setting enables us to specify a ground truth that allows us to empirically assess the performance of the policies learned by PPO. In this analysis, we consider different values for the basic reproductive number R_0 and the population composition (i.e., proportion of adults, children, elderly, adolescents) of the district. Both parameters induce a significant change of the epidemic model's dynamics. Through these experiments, we demonstrate the potential of deep reinforcement learning algorithms to learn policies in the context of complex epidemiological models, opening the prospect to learn in even more complex stochastic models with large action spaces. In this regard, we consider a large scale setting where we examine whether there is an advantage to consider the collaboration between districts when designing school closure policies.

To situate this work in the state-of-the-art, we note that the concept to learn dynamic policies by formulating the decision problem as a Markov decision process (MDP) was introduced in [2]. To our best knowledge, the work presented in this manuscript is the first attempt to use deep reinforcement learning algorithms directly on a complex meta-population model.

To summarize, in this work, we demonstrate the potential of deep reinforcement learning in the context of complex stochastic epidemiological models. As few assumptions are made on the epidemiological model, our new technique has the potential to be used for other epidemiological settings, such as the ongoing COVID-19 pandemic. For future work, it would be interesting to investigate how well these algorithms scale to even larger state and/or action spaces.

References

1. T. C. Germann, H. Gao, M. Gambhir, A. Plummer, M. Biggerstaff, C. Reed, and A. Uzicanin. School dismissal as a pandemic influenza response: When, where and for how long? *Epidemics*, page 100348, 2019.
2. R. Yaesoubi and T. Cohen. Dynamic health policies for controlling the spread of emerging infections: influenza as an example. *PloS one*, 6(9), 2011.

Generalized Optimistic Q-Learning with Provable Efficiency*

Grigory Neustroev[1][0000−0002−7706−7778] and
Mathijs M. de Weerdt[1][0000−0002−0470−6241]

Delft University of Technology, EEMCS, Algorithmics,
P.O. Box 5031, 2600 GA, Delft, the Netherlands
{g.neustroev, m.m.deweerdt}@tudelft.nl

This is an abstract of a paper published at the 19th International Conference on Autonomous Agents and Multiagent Systems, Auckland, New Zealand [2].

1 Introduction

When a learning algorithm requires as few data samples as possible, it is called sample efficient. Recently, Jin *et al.* introduced the first provably efficient model-free reinforcement learning (RL) algorithm [1]. Later, a few other sample-efficient model-free algorithms were developed [4, 3]. The key factor that allows these algorithms to achieve sample efficiency is their use of the principle of *optimism in the face of uncertainty*.

The paper studies the effect of optimism on sample efficiency of RL. It presents a generalized theory on optimistic model-free RL, unifying the existing algorithms. Using this theory, we establish sample efficiency of optimistic Q-learning by showing that its regret grows sub-linearly with respect to the number of samples. Moreover, we show that the regret of optimistic Q-learning can be explained by three distinct factors.

2 Generalized Optimistic Q-Learning

In learning, optimism is used in two ways: optimistic initialization and optimistic exploration. We look at the existing optimistic model-free RL methods [1, 4, 3] to see how they incorporate these aspects of optimism.

In initialization, large values are assigned to all state-action combinations. This guarantees that actions never chosen before seem especially lucrative. When such initialization is not possible (e.g., in deep RL), the Q-values of unvisited states are augmented with a bonus term that we call a *bonus for optimism*.

Optimistic exploration is done by using upper confidence bounds (UCBs) on state-action values. In each interaction, the action with the highest UCB is chosen. This happens either if the true optimal value is high, or if there is not enough confidence in it yet. In the former case, the agent essentially performs

* This research received funding from the Netherlands Organization for Scientific Research (NWO).

exploitation, as the chosen action is the best one. The latter case represents exploration, because an action with high uncertainty in its outcome is chosen. Thus, UCBs help to automatically balance exploration and exploitation. To maintain the UCBs, a *confidence bonus* is added to the Q-values during learning.

We incorporate these bonuses in an algorithm that we call *generalized optimistic Q-learning* and perform a theoretical analysis of its sample efficiency. Unlike previous results, our analysis does not rely on the particular form of the bonuses to determine whether the resulting algorithm is sample efficient or not.

The general form we use allows us to show that the total regret R of optimistic Q-learning is asymptotically bounded by the sum of three different terms:

$$R = \mathrm{O}\big(\mu(X + B + E)\big). \tag{1}$$

The state-action space size X represents the effect of the *optimistic initialization*, as the number of initial values is equal to X. The bonus effect B depends on the bonuses for optimism and for confidence. The last term E represents the required number of interactions with the environment to ensure that all of the possible outcomes are experienced with high probability. The magnitude μ depends on the reward range and the discounting factor and represents the scale of Q-value.

The formal proof of this regret bound relies on some mild necessary conditions. They can be found in the paper along with the formal definitions of the terms μ, X, B, and E and the proof itself. The paper also gives an example of a new algorithm designed within the generalized optimistic Q-learning framework. This algorithm, called UCB-H$^+$, is similar to UCB-H [1], but uses a different learning rate. Using the theoretical framework of the paper, we prove that it is sample-efficient. Then we evaluate UCB-H$^+$ in two experiments, which demonstrate a regret reduction of 13% and 43% compared to UCB-H.

3 Conclusions

Generalized optimistic Q-learning incorporates existing optimistic model-free reinforcement learning, and our proof does not rely on a particular form of learning rate or bonuses, allowing transfer of these results to new algorithms.

References

1. Jin, C., Allen-Zhu, Z., Bubeck, S., Jordan, M.I.: Is Q-learning provably efficient? In: Advances in Neural Information Processing Systems 31, pp. 4863–4873. Curran Associates, Inc. (2018)
2. Neustroev, G., de Weerdt, M.M.: Generalized optimistic Q-learning with provable efficiency. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. pp. 913–921 (2020)
3. Rashid, T., Peng, B., Boehmer, W., Whiteson, S.: Optimistic exploration even with a pessimistic initialisation. In: International Conference on Learning Representations. Addis Ababa, Ethiopia (Apr 2020)
4. Wang, Y., Dong, K., Chen, X., Wang, L.: Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. In: International Conference on Learning Representations. Addis Ababa, Ethiopia (Apr 2020)

From Continuous Observations to Symbolic Concepts: A Discrimination-Based Strategy for Grounded Concept Learning

Jens Nevens, Paul Van Eecke, and Katrien Beuls

Artificial Intelligence Lab, Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussels, Belgium
{jens|paul|katrien}@ai.vub.ac.be

In order to reason and communicate about their environment, autonomous agents need to be able to distill meaningful concepts from the observed streams of continuous sensori-motor data. In this paper, we report on computational simulations of how such concepts are distilled through a series of situated communicative interactions. Our approach builds further on earlier work within the language game paradigm [5], where concepts were either limited to continuous data on a single feature channel (e.g. [1]) or to non-continuous data on multiple feature channels (e.g. [6]). We lift both restrictions at the same time. Through a tutor-learner scenario, our novel method allows an agent to construct meaningful concepts which are formed by discriminative combinations of prototypical values on human-interpretable feature channels. Most current approaches that bridge between the continuous and symbolic domain make use of deep learning techniques (e.g. [2]). These approaches often achieve high levels of accuracy but they rely on large amounts of training data, the resulting models lack transparency and they require partial or complete re-training to accommodate changes in the environment.

The experiments are set in an environment based on the CLEVR dataset [4]. This environment consists of scenes with geometrical objects of different colours, shapes, sizes and materials. In each interaction, the tutor uses a single word to refer to one of the objects, e.g. *"sphere"*. The learner observes the scene through continuous-valued and human-interpretable feature channels, such as 'area', 'number-of-corners' or 'width-height-ratio'. These features are obtained through simulation (simulated world setting) or object detection, segmentation and feature extraction techniques [3] (noisy world setting). The task of the learner is to point out the object meant by the tutor. It does so by computing the similarity between each object and the current representation of the concept that is associated with the word form uttered by the tutor. At the end of the interaction, the learner receives feedback on whether or not it was correct and the tutor points out the correct object. Using this information, the learner can update the concept it used. More specifically, the learner rewards the most discriminating subset of feature channels and punishes the others. Additionally, all prototypical values are shifted slightly towards the object. For each concept, the learner must simultaneously learn which feature channels are important and what their prototypical values should be. Figure 1 shows the communicative success of the agents and an example of a learned concept.

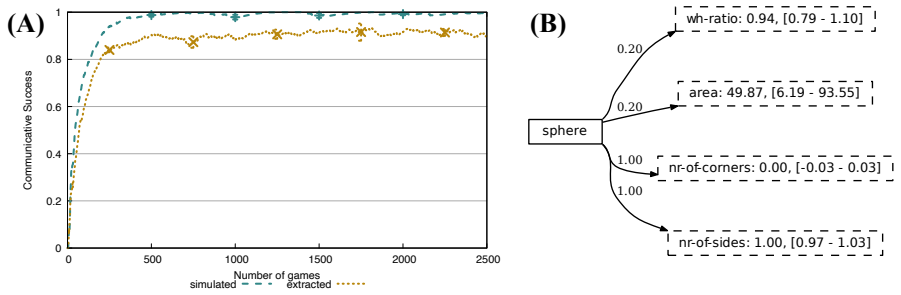


Fig. 1. (A) The agent achieves 100% communicative success in the simulated world and 91% in the noisy world **(B)** Concepts are represented through a weighted set of attributes. The weight corresponds to the certainty of an attribute belonging to the concept. Each attribute is modelled as a normal distribution that keeps track of its prototypical value (the mean) and the standard deviation. The weighted sets capture discriminative combinations of attributes. The concept SPHERE focusses on attributes related to shape.

Through a range of experiments, we showcase several desirable properties of our approach. The first experiment shows that the agent rapidly adapts to changes in the environment and the approach allows for incremental learning. In the second experiment, we demonstrate that the concepts generalise well to unseen settings. Finally, we show that the concepts can be learned even when combined compositionally. These properties, combined with fast and data-efficient learning and human-interpretable representations, make our approach well-suited to be used in robotic agents for mapping continuous sensory input to grounded, symbolic concepts. These can in turn be used for higher-level reasoning tasks, such as navigation, (visual) question answering and action planning.

References

1. Bleys, J.: Language strategies for the domain of colour. Language Science Press, Berlin (2015)
2. Dolgikh, S.: Spontaneous concept learning with deep autoencoder. International Journal of Computational Intelligence Systems **12**(1), 1–12 (2018)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969. Honolulu, Hawaii (2017)
4. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2901–2910. Honolulu, Hawaii (2017)
5. Steels, L.: Language games for autonomous robots. IEEE Intelligent systems **16**(5), 16–22 (2001)
6. Wellens, P.: Coping with combinatorial uncertainty in word learning: A flexible usage-based model. In: The Evolution Of Language, pp. 370–377. World Scientific (2008)

Learning 2-opt Local Search for the Traveling Salesman Problem

Paulo R. de O. da Costa, Jason Rhuggenaath, Yingqian Zhang, and Alp Akcay

Eindhoven University of Technology, 5612 AZ Eindhoven, Netherlands
{p.r.d.oliveira.da.costa, j.s.rhuggenaath, yqzhang, a.e.akcay}@tue.nl

Abstract. Recent works using deep learning to solve the Traveling Salesman Problem (TSP) have focused on learning construction heuristics. Such approaches require additional procedures such as beam search and sampling to improve solutions and achieve state-of-the-art performance. However, few studies have focused on improvement heuristics, where a given solution is improved until reaching a near-optimal one. In this work, we propose to learn a local search heuristic based on 2-opt operators via deep reinforcement learning. We propose a policy gradient algorithm to learn a stochastic policy that selects 2-opt operations given a current solution. Moreover, we introduce a policy neural network that leverages a pointing attention mechanism, which unlike previous works, can be easily extended to more general k-opt moves. Our results show that the learned policies can improve even over random initial solutions and approach near-optimal solutions at a faster rate than previous state-of-the-art deep learning methods.

Keywords: Deep Reinforcement Learning · Combinatorial Optimization · Traveling Salesman Problem.
Acknowledgments: This research is funded by NWO Big data: Real Time ICT for Logistics, project number 628.009.012
Publication: The full paper of this abstract has been accepted at the 12th Asian Conference on Machine Learning (ACML), 2020. Available at http://proceedings.mlr.press/v129/costa20a/costa20a.pdf

1 Introduction

The Traveling Salesman Problem (TSP) is a well-known NP-hard combinatorial optimization problem. Exact methods for the TSP such as linear programming [1] are guaranteed to find an optimal solution but are often too expensive computationally. On the other hand, designed heuristics require specialized knowledge and their performances are often limited by algorithmic design decisions.

Thus, a machine learning method could potentially learn better heuristics by extracting useful information directly from data. We focus on methods in which a given solution is improved sequentially until reaching an (local) optimum. Thus, we propose a deep reinforcement learning algorithm to learn improvement

heuristics based on 2-opt moves. Our approach can achieve near-optimal results that are better than previous deep learning methods based on construction and improvement heuristics.

2 Methods

Our neural network follows the general encoder-decoder architecture. The encoder embeds both graph topology and the positions of each node in a solution. Given node and sequence embeddings the *policy* decoder is autoregressive and samples output actions one element at a time. The *value* decoder operates on the same representations but generates real-valued outputs to estimate state values.

In our formulation, we resort to the Policy Gradient learning rule, to optimize our policy. Our model is close to REINFORCE [3] but with periodic episode length updates. Thus, at the start the agent learns how to behave over small episodes for easier credit assignment, later tweaking its policy over larger horizons.

3 Results

We learn policies for TSP instances with 20, 50 and 100 nodes, and depict the optimality gap for 10,000 test instances in Table 1. The results show that we can learn effective policies that decrease the optimality gap over the training epochs and can outperform the effective Graph Attention (GAT) [2] and are close to the optimal solutions.

Table 1: Performance of TSP methods w.r.t. Concorde. *Type*: **RL**: Reinforcement Learning, **S**: Sampling, *Time*: Time to solve 10,000 instances.

Method	Type	TSP20			TSP50			TSP100		
		Cost	Gap	Time	Cost	Gap	Time	Cost	Gap	Time
Concorde [1]	Solver	3.84	0.00%	(1m)	5.70	0.00%	(2m)	7.76	0.00%	(3m)
GAT [2]	RL,S	3.84	0.08%	(5m)	5.73	0.52%	(24m)	7.94	2.26%	(1h)
Ours	RL	**3.84**	**0.00**%	(15m)	5.70	0.12%	(29m)	**7.83**	**0.87**%	(41m)

References

1. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The traveling salesman problem: a computational study. Princeton university press (2006)
2. Kool, W., van Hoof, H., Welling, M.: Attention, learn to solve routing problems! In: Proceedings of the 7th International Conference on Learning Representations (ICLR) (2019)
3. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4), 229–256 (1992)

Recent Advances in Multi-Objective Multi-Agent Decision Making

Roxana Rădulescu[1], Patrick Mannion[2], Diederik M. Roijers[1,3], and Ann Nowé[1]

[1] Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium
[2] School of Computer Science, National University of Ireland Galway, Ireland
[3] Microsystems Technology, HU Univ. of Applied Sciences Utrecht, the Netherlands

Numerous real-world problems involve both multiple actors and objectives that should be taken into account when making a decision. Multi-objective multi-agent systems (MOMAS) represent an ideal setting to study such problems, but given the increasingly complex dimensions involved, it still remains an understudied domain despite its high relevance. We present here a short overview of our recent advances in multi-objective multi-agent decision making settings.

MOMAS Taxonomy In MOMAS the reward signal for each agent is a vector, where each component represents the performance on a different objective. We consider that compromises between competing objectives should be made on the basis of the utility that these compromises have for the users. In other words, we assume there exists a utility function that maps the vector value of a compromise solution to a scalar utility.

In order to offer a unified view of the field, we build a taxonomy (Figure 1) of what constitutes a solution for a multi-objective multi-agent decision problem based on reward and utility functions. More details on each setting and solution concept can be found in [2].

		Utility: Team	Utility: Social choice	Utility: Individual
Reward: Team		Coverage sets	Mechanism design	Coverage sets (+ Negotiation) Equilibria & stability concepts
Reward: Individual			Mechanism design	Equilibria & stability concepts Coverage Sets as best responses

Fig. 1: Multi-objective multi-agent decision making taxonomy and mapping of solution concepts.

Another factor we identify is the difference between the optimisation criteria: expected scalarised returns (ESR) and scalarised expected returns (SER) [1].

This roughly distinguishes settings where either the utility of a single outcome (ESR) or the utility of the average outcome over multiple runs (SER) matters.

Learning in MONFGs We have studied multi-objective normal form games under the SER optimisation criterion with non-linear utility functions [3]. We show by example that while Nash equilibria (NE) need not exist, correlated equilibria (CE) can still be present when optimising with respect to a single given signal (i.e., single-signal CE).

Opponent modelling in MONFGs When the same multi-objective reward vector leads to different utilities for each user, it becomes essential for an agent to learn about the behaviour of other agents in the system. In [4] we present the first study of the effects of opponent modelling (OM) on MONFGs with non-linear utilities, under the SER criterion. We demonstrate that OM can alter the learning dynamics in this setting: when there are no NE, OM can have adverse effects on utility, or a neutral effect at best; when equilibria are present, OM can confer significant benefits (Figure 2).

	L	M	R
L	$(4,1)$	$(1,2)$	$(2,1)$
M	$(3,1)$	$(3,2)$	$(1,2)$
R	$(1,2)$	$(2,1)$	$(1,3)$

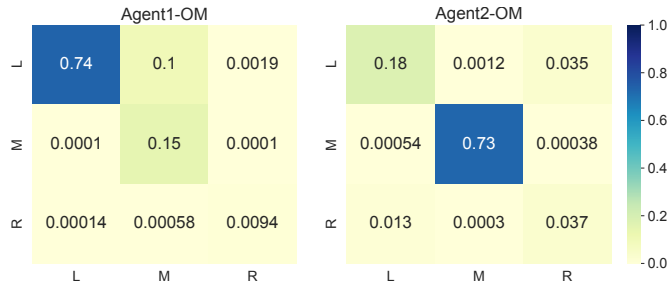


Fig. 2: Empirical outcome distributions when agents are using opponent modelling with utility functions $u_1(\mathbf{p}) = p^1 \cdot p^1 + p^2 \cdot p^2$ and $u_2(\mathbf{p}) = p^1 \cdot p^2$. Opponent modelling allows each agent to steer the outcome towards its preferred NE. Agent 1 obtains the highest SER under (L,L), while for Agent 2 that is (M, M).

References

1. Roijers, D.M., Steckelmacher, D., Nowé, A.: Multi-objective reinforcement learning for the expected utility of the return. In: Proceedings of the Adaptive and Learning Agents workshop at FAIM. vol. 2018 (2018)
2. Rădulescu, R., Mannion, P., Roijers, D.M., Nowé, A.: Multi-objective multi-agent decision making: a utility-based analysis and survey. Autonomous Agents and Multi-Agent Systems **34** (2020)
3. Rădulescu, R., Mannion, P., Zhang, Y., Roijers, D.M., Nowé, A.: A utility-based analysis of equilibria in multi-objective normal-form games. The Knowledge Engineering Review **35**, e32 (2020)
4. Zhang, Y., Rădulescu, R., Mannion, P., Roijers, D., Nowé, A.: Opponent modelling for reinforcement learning in multi-objective normal form games. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. pp. 2080–2082 (2020)

Multi-Agent Thompson Sampling for Bandits with Sparse Neighbourhood Structures

Timothy Verstraeten[1], Eugenio Bargiacchi[1], Pieter J.K. Libin[1], Jan Helsen[1], Diederik M. Roijers[1,2], and Ann Nowé[1]

[1] Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
[2] HU University of Applied Sciences Utrecht, Utrecht, 3584CS, the Netherlands

This is an extended abstract of the article that was accepted and published in Nature Scientific Reports: https://doi.org/10.1038/s41598-020-62939-3

Multi-agent coordination is prevalent in many real-world applications, such as traffic light control, warehouse commissioning and wind farm control. Often, such settings can be formulated as coordination problems in which agents have to cooperate in order to optimize a shared team reward. Handling multi-agent settings is challenging, as the size of the joint action space scales exponentially with the number of agents in the system. Therefore, an approach that directly considers all agents' actions jointly is computationally intractable. This has made such coordination problems the central focus in the planning literature. Fortunately, in real-world settings agents often only directly affect a limited set of neighbouring agents. This means that the global reward received by all agents can be decomposed into local components that only depend on small subsets of agents. Exploiting such loose couplings is key in order to keep multi-agent decision problems tractable.

In this work, we consider learning to coordinate in multi-agent systems. While most of the literature only considers approximate reinforcement learning methods for learning in multi-agent systems, it has recently been shown that it is possible to achieve theoretical bounds on the regret (i.e., how much reward is lost due to learning). In this work, we use the multi-agent multi-armed bandit problem definition, and improve upon the state of the art. Specifically, we propose the multi-agent Thompson sampling (MATS) algorithm [5, 6], which exploits loosely-coupled interactions in multi-agent systems.[3] The loose couplings are formalized as a *coordination graph*, which defines for subsets of agents whether their actions depend on each other. We assume the graph structure is known beforehand, which is the case in many real-world applications with sparse agent interactions (e.g., wind farm control). Additionally, our method leverages the exploration-exploitation mechanism of Thompson sampling (TS). TS has been shown to be highly competitive to other popular methods, e.g., the Upper Confidence Bound algorithm [3]. Recently, theoretical guarantees on its regret have been established, which renders the method increasingly popular in the literature. Additionally, due to its Bayesian nature, problem-specific priors

[3] The source code of MATS is available at github.com/Svalorzen/AI-Toolbox. [1]

can be specified, which has strong relevance in many practical fields, such as advertisement selection and influenza mitigation.

We provide a finite-time Bayesian regret analysis and prove that the upper regret bound of MATS is low-order polynomial in the number of actions of a single agent for sparse coordination graphs. This is a significant improvement over the exponential bound of classic TS, which is obtained when the coordination graph is ignored. Moreover, we show that MATS improves upon the state-of-the-art algorithms, Multi-Agent Upper Confidence Exploration (MAUCE) [2] and Sparse Cooperative Q-Learning (SCQL) [4], in various synthetic settings. Although MATS and MAUCE have similar theoretical guarantees, we found that MATS consistently outperforms both MAUCE and SCQL empirically. We argue that the high performance of MATS is due to the ability to seamlessly include domain knowledge about the reward distributions and treat the problem parameters as unknowns. To highlight the power of this property, we introduced a novel setting with skewed reward distributions. As MAUCE only supports symmetric exploration bounds, it is challenging to correctly assess the amount of exploration needed to solve this task. In contrast, MATS has the ability to exploit the shape of the reward distribution to achieve more targeted exploration. Finally, we demonstrate the practical benefits of MATS on a realistic wind farm control task. As wind passes through the farm, downstream turbines observe a significantly lower wind speed. This is known as the *wake effect*, which is due to the turbulence generated behind operational turbines. Wake redirection is a control mechanism where turbines' rotors are misaligned to deflect wake away from the wind farm. While a misaligned turbine produces less energy on its own, the group's total productivity is increased. Physically, the wake effect reduces over long distances, and thus, turbines tend to only influence their neighbours. We can use this domain knowledge to define groups of agents and organize them in a graph structure. We demonstrate that MATS achieves state-of-the-art performance on the wind farm control task.

References

1. Bargiacchi, E., Roijers, D.M., Nowé, A.: AI-Toolbox: A C++ library for reinforcement learning and planning (with python bindings). Journal of Machine Learning Research **21**(102), 1–12 (2020), http://jmlr.org/papers/v21/18-402.html
2. Bargiacchi, E., Verstraeten, T., Roijers, D., Nowé, A., van Hasselt, H.: Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In: International Conference on Machine Learning (2018)
3. Chapelle, O., Li, L.: An empirical evaluation of Thompson sampling. In: Advances in Neural Information Processing Systems (NIPS). vol. 24, pp. 2249–2257 (2011)
4. Kok, J.R., Vlassis, N.: Sparse cooperative q-learning. In: Proc. of the 21st International Conference on Machine Learning. New York, NY, USA (2004)
5. Verstraeten, T., Bargiacchi, E., Libin, P.J.K., Helsen, J., Roijers, D.M., Nowé, A.: Multi-agent Thompson sampling for bandit applications with sparse neighbourhood structures. Sci. Rep. **10** (2020). https://doi.org/10.1038/s41598-020-62939-3
6. Verstraeten, T., Bargiacchi, E., Libin, P.J.K., Roijers, D.M., Nowé, A.: Thompson sampling for factored multi-agent bandits. In: Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems. pp. 2029–2031 (2020)

Demonstrations

Communication Training in Virtual Reality: A Training Application for the Dutch Railways

Eric Jutten[1], Edward Bosma[1], Kiki Buijs[1], Romy Blankendaal[2] and Tibor Bosse[3]

[1] The Simulation Crew, Nijmegen, The Netherlands
[2] TNO, Soesterberg, The Netherlands
[3] Radboud University, Nijmegen, The Netherlands
eric.jutten@simulationcrew.com

1 Introduction

Virtual reality (VR) provides many promising opportunities with regard to training of communication skills, as it provides a medium where users can safely practice their skills by engaging in social interactions with Intelligent Virtual Agents through verbal and non-verbal modalities [1]. This way, VR allows users to practice what to say when, for instance to learn how to respond to a nervous interlocutor, or how to communicate professionally under increasing levels of stress. The ability for people to practice their skills is of great importance, as the transfer of 'knowing how to do something' to 'actually doing it' is difficult, even when one knows what he/she is expected to do [2].

In 2019, The Simulation Crew has developed an interactive VR communication trainer that allows users to interact with Intelligent Virtual Agents through speech. To process the user's input and generate appropriate output, the system makes use of a number of AI techniques, including speech recognition, multi-modal social signal analysis, and dialog modelling. These features allow the user to interact with the system using free speech, which distinguish it from many communication trainers that based on multiple choice menus (e.g., [1]). As a result, users have a more natural experience while interacting with the virtual agents. Hence, users are encouraged to be more actively involved with the material and come up with their own input, which may result in better outcomes after training [3]. In addition to the verbal communication, this application also takes into account various aspects of the non-verbal communication of the user, such as prosody and gaze direction. It can thus provide the user feedback on the effect of non-verbal behaviours such as nodding and humming on the interlocutor.

2 Application

The NS (the largest passenger rail transport company in the Netherlands), is highly interested in the use of a VR communication training for their personnel and willing to cooperate in scientific research. Within NS, education and training of their personnel is of great importance. Traditionally, this is often done by making use of role play. How-

ever, employees sometimes feel uncomfortable when participating in role play, especially with other people present and watching them while practicing. They therefore encounter the feeling of being assessed instead of a safe environment to practice.

Within this new VR communication training, employees are instructed to help a disabled passenger to get on the train. In this scenario, employees can practice with more visual tasks like situating the gangway, but also with social tasks related to communication with passengers. This pilot consists of several use cases, for instance allowing users to practice in a quiet scenario, or in scenarios that include several stress factors.

Users typically play the scenarios twice with the opportunity to make mistakes, repeat the actions and learn. They are also provided the opportunity to intentionally make mistakes in order to see the effect on the development of the scenario. The effect of the users' actions on the development of the scenario provides users with feedback during their training. An example of this type of feedback is the verbal and non-verbal response of the passenger (e.g., when the user ignores a request, the passenger may become upset). In addition, at the end of the scenario users receive feedback on their performance, both by the passenger and by the system.

The training concludes with a scenario where the user encounters several stress factors. Within this scenario, users will be distracted from their original task in order to challenge them to keep communicating professionally when encountering distractions that resemble distractions in real-life.

3 Conclusion

In August 2020, a pilot evaluation of the training system has been conducted in collaboration with Radboud University and TNO. During three sessions, a total of 30 employees of NS have tested the system, and both qualitative and quantitative feedback was gathered. Our preliminary conclusion is that participants were generally very positive about the system, and even though they signalled various points for improvement they were particularly enthusiastic about the ability to engage with the virtual agents in an 'open' conversation using free speech. Furthermore, they confirmed that this type of training offers promising possibilities for training their personnel, in particular for new employees. A working demonstration of the system will be presented at the conference.

References

1. Blankendaal, R.A.M., Gerritsen, C., Otte, M., & Bosse, T. (2018). A virtual reality application for aggression de-escalation training in public transport. In: 30th Benelux Conference on Artificial Intelligence, BNAIC 2018 (pp. 5-20).
2. Kahlke, R.M., McConnell, M.M., Wisener, K.M., and Eva, K.W. (2020). The disconnect between knowing and doing in health professions education and practice. Adv Heal Sci Educ. 2020; 25(1):227-240. doi:10.1007/s10459-019-09886-5.
3. Deslauriers L, McCarty LS, Miller K, Callaghan K, Kestin G. (2019). Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. Proc Natl Acad Sci USA. 2019; 116(39):19251-19257. doi:10.1073/pnas.1821936116.

A Multifunctional, Interactive DMN Decision Modelling Tool

Simon Vandevelde* and Joost Vennekens

KU Leuven, De Nayer Campus, Dept. of Computer Science
Leuven.AI - KU Leuven Institute for AI, B-3000 Leuven, Belgium
{s.vandevelde, joost.vennekens}@kuleuven.be

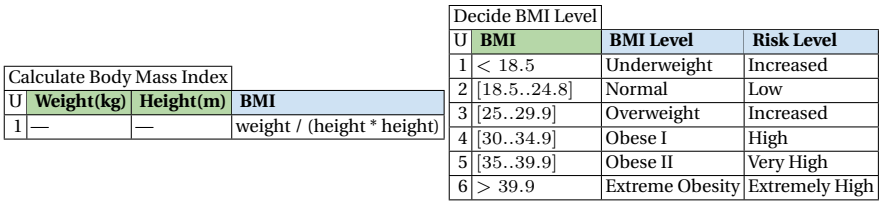
Abstract. In this demo we showcase DMN-IDP, a user-friendly tool which combines the readability of the Decision Model and Notation (DMN) standard with the power of the IDP system through an interactive interface.

The Decision Model and Notation (DMN) standard is a table-based way of representing decision logic, with a focus on readability and user-friendliness. Designed by the Object Management Group, it was quickly adopted in various industries. In academia, interest in DMN to represent knowledge is also growing, because of its accessibility as a modelling language for domain experts [6]. To use DMN models, tools exist which can compute a suitable assignment of values to the decision variables, given the values of the environmental variables, by means of forward propagation.

In [4], it was argued that the knowledge expressed in a DMN model can be used for much more. For instance, value propagation can also be done in other directions, such as from decision to environmental variables. Other examples are reasoning on incomplete data, and applying different inference tasks, such as optimization. To illustrate their approach, the authors made use of the IDP knowledge base system [5]. By manually translating DMN models into first-order logic knowledge bases (KBs), users could interact with the KB in a user-friendly way via a browser-based interface. While this results in a powerful and flexible way of working, there are two main downsides. Firstly, the DMN models need to be created in a separate tool. Secondly, the translation from DMN to IDP KB is done manually, for which knowledge of the representation language of the IDP system is required.

In this demo, we present DMN-IDP, a full-fledged DMN tool which combines the dmn-js DMN editor [2] and the IDP-based Interactive Consultant interface [3]. Using this tool, a user can upload or create DMN models, which are then translated into IDP KBs. Users can interact with these models via the Interactive Consultant interface. The translation from DMN to IDP is done by the same transformation used in the cDMN framework [1]. The interface supports propagating values in any direction, reasoning on incomplete data, optimization of values and explanation of decisions. In this way, DMN models become useable in more situations, removing the need to build specific models for every target output in a use case.

* This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.



Calculate Body Mass Index

U	Weight(kg)	Height(m)	BMI
1	—	—	weight / (height * height)

Decide BMI Level

U	BMI	BMI Level	Risk Level
1	< 18.5	Underweight	Increased
2	[18.5..24.8]	Normal	Low
3	[25..29.9]	Overweight	Increased
4	[30..34.9]	Obese I	High
5	[35..39.9]	Obese II	Very High
6	> 39.9	Extreme Obesity	Extremely High

Fig. 1: A DMN model for deciding a patient's BMI level and risk level.

As an example, consider the DMN model in Figure 1, which calculates a BMI level and risk level based on a patient's weight and height. The table on the left consists of one rule, which is read as "For every possible weight and height, the BMI is weight divided by height squared." The table on the right then decides what the value for *BMI Level* and *Risk Level* is, based on the BMI. Using this model, standard DMN tools could for example calculate that for a weight of 100kg and height of 1.8m, the BMI is 30.9, resulting in a high risk level. However, say we now want to know the opposite, i.e., what weight would give a 1.8m patient a low risk level. Standard DMN tools cannot infer this information from this model. Our tool on the other hand is capable of reasoning backwards, even with incomplete data. This allows us to enter the height and set the value of *Risk Level* to "Low" while leaving the weight unknown. By now maximizing the *Weight* variable, we find that a weight less than 80.7kg results in a low risk for the height.

The tool also includes some basic functionality for detecting common errors in DMN specifications. We plan to develop this further in future work, along with a functionality to improve the traceability of decisions.

During the demo, participants will get to interact with our tool via multiple use cases, allowing them to explore the capabilities of the system freely. They will also be encouraged to experiment with the DMN models themselves, so that they can learn the connections between the components. An online version of the tool is available at https://autoconfig-dmn.herokuapp.com/.

References

1. Aerts, B., Vandevelde, S., Vennekens, J.: Tackling the dmn challenges with cdmn: A tight integration of dmn and constraint reasoning. Springer (2020)
2. bpmn.io: Dmn viewer and editor. https://bpmn.io/toolkit/dmn-js/ (2015)
3. Carbonnelle, P., Aerts, B., Deryck, M., Vennekens, J., Denecker, M.: An interactive consultant. In: Proceedings of the 31st Benelux Conference on Artificial Intelligence (BNAIC 2019) and the 28th Belgian Dutch Conference on Machine Learning (Benelearn 2019), Brussels, Belgium, November 6-8, 2019 (2019)
4. Dasseville, I., Janssens, L., Janssens, G., Vanthienen, J., Denecker, M.: Combining dmn and the knowledge base paradigm for flexible decision enactment (2016)
5. De Cat, B., Bogaerts, B., Bruynooghe, M., Janssens, G., Denecker, M.: Predicate logic as a modeling language: The idp system. pp. 279–329. ACM Books (2018)
6. Deryck, M., Hasic, F., Vanthienen, J., Vennekens, J.: A case based inquiry into the decision model and notation (dmn) and the knowledge base (kb) paradigm. In: Proceedings of RuleML+RR 2018. vol. 11092 LNCS, pp. 248–263. Springer (2018)

Thesis abstracts

Learning What to Attend to: Using Bisimulation Metrics to Explore and Improve Upon What a Deep Reinforcement Learning Agent Learns*

Nele Albers, Miguel Suau de Castro, and Frans A. Oliehoek

Delft University of Technology, Delft, Netherlands
{N.Albers, M.SuauDeCastro, F.A.Oliehoek}@tudelft.nl

Keywords: Deep Reinforcement Learning · Representation Learning · Bisimulation Metrics · Markovianity.

Recent years have seen a surge of algorithms and architectures for deep Reinforcement Learning (RL), many of which have shown remarkable success for various problems. Yet, little work has attempted to relate the performance of these algorithms and architectures to what the resulting deep RL agents actually learn, and whether this corresponds to what they should ideally learn. Such a comparison may allow for both an improved understanding of why certain algorithms or network architectures perform better than others and the development of methods that specifically address discrepancies between what is and what should be learned.

Ideal Representation. The concept of ideal representation we utilize is the Coarsest Markov State Representation (CMSR). We define this representation as one in which the Euclidean distances between states are proportional to how "behaviorally different" [2] those states are. Behavioral similarity thereby is measured by a specific bisimulation metric [1]. This bisimulation metric regards states as equivalent if and only if they have the same expected reward and transition distribution over all state equivalence classes for all actions. Moreover, if the parameters of two equivalent states are altered on a small scale, the metric distance between the states will stay small. Learning an internal state representation that is similar to the CMSR has several desirable theoretical properties:

- The CMSR is the smallest state representation that still allows for the prediction of the reward and next state [3].
- The CMSR does not distinguish states based on features that are irrelevant for predicting the next reward and internal state. Thus, a policy learned based on this representation generalizes to different values for such features.
- If a subset of the features required for predicting the reward and next internal state for a domain is sufficient for predicting the reward and next internal state after modifying the reward or the transition function, the CMSR for the original domain suffices to learn the Q-values of a thus modified domain.

* Full thesis available at http://resolver.tudelft.nl/uuid:2945dcc8-e7b9-4536-b9e7-074cfe86d3f9.

– Making the Euclidean distances between internal states proportional to their behavioral similarity renders the formed representation less sensitive to small estimation errors if the transition or reward functions are approximated.

Research Objective. It is hence *theoretically* desirable that deep RL agents learn the CMSR. Yet, we do not know to which extent deep RL agents learn the CMSR, and whether doing so is useful *in practice*. Thus, we look at the internal state representations learned by deep RL agents at various stages during training and under different training conditions, and compare them to the CMSR. Furthermore, to elucidate the practical usefulness of learning the CMSR, we contrast the learning speeds and consistencies and the generalization performances of neural networks with hidden-layer representations that differ in how similar to the CMSR they are, while controlling for other factors.

Contributions. We split our contributions into *methodological* and *experimental* ones. Our methodological contributions are as follows:

– We propose using correlation coefficients based on bisimulation metrics to measure how similar to the CMSR an internal state representation is. These correlation coefficients also allow to specifically determine whether an internal state representation is Markov with respect to the rewards or Markov with respect to the transitions[1].
– We introduce an auxiliary loss that pushes a neural network to learn an internal state representation that is similar to the CMSR in a network layer.

We further provide experimental contributions:

– We identify three overlapping learning phases that together make up the learning process of deep RL agents using model-free Q-learning agents as example. Thereby, it is during the second learning phase that internal state representations become increasingly similar to the CMSR. We also point out several factors that impact this learning process. The precise CMSR is not learned in any of our experiments.
– We show that learning a hidden-layer representation that is more similar to the CMSR *during* training can speed up the learning process and cause good solutions to be found more reliably.
– We demonstrate that learning a hidden-layer representation that is more similar to the CMSR *by the end of* training may lead to improved generalization to new irrelevant feature values. Creating such a representation also may enable better generalization to related domains with modified reward or transition functions, as long as the modifications do not render formerly irrelevant features relevant.

[1] A state representation that is *Markov with respect to the reward* is one in which knowledge of previous internal states does not lead to a more accurate prediction of the next reward [4]. The definition of *Markov with respect to the transition* proceeds analogously.

Acknowledgments. This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

References

1. Ferns, N., Panangaden, P., Precup, D.: Metrics for finite markov decision processes. In: Proceedings of the 20th conference on Uncertainty in artificial intelligence. pp. 162–169. AUAI Press (2004)
2. Ferns, N., Precup, D.: Bisimulation metrics are optimal value functions. In: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence. pp. 210–219 (2014)
3. Givan, R., Dean, T., Greig, M.: Equivalence notions and model minimization in markov decision processes. Artificial Intelligence **147**(1-2), 163–223 (2003)
4. McCallum, R.: Reinforcement learning with selective perception and hidden state (1997)

Capturing Implicit Biases With Positive Operators

J. Bosscher[1,2], *Supervisors:* dr. M.A.F. Lewis[1,2], and dr. K. Schulz[1,2]

[1] University of Amsterdam, Science Park 904, 1012 WX Amsterdam, The Netherlands
[2] Institute for Logic, Language and Computation, Science Park 107, 1098 XG Amsterdam, The Netherlands
j.m.bosscher@uva.nl

Keywords: positive operator · hyponymy · graded hyponymy · distributional semantics · implicit bias · bias · stereotypes · word embeddings

1 Introduction

Modelling words as vectors has been an extremely successful way of representing word meaning in a manner that can be programmed into a computer [5] [7][9]. Word vectors have been shown to be affected by the ideas and beliefs of the humans that generated the corpora they are extracted from [1] [10]. Caliskan et. al. [2] showed that our biases and stereotypical beliefs could also be extracted from these representations. In order to identify implicit biases held by human speakers, the results from a physiological test called the Implicit Association Test (IAT) [4] are used as a benchmark. Caliskan et. al. [2] then compared these results to that of their Natural Language Processing (NLP) version of the IAT called the Word Embedding Association Test (WEAT). Utilizing widely adopted distributional models, mainly focusing on GloVe embeddings [9], they were able to replicate every association documented by the IAT that they tested. This leads them to expect that human biases are in general retrievable from statistical properties of language use.

However, the two test methods used to extract implicit biases from human speakers on the one hand and from corpora on the other differ clearly in methodology. The IAT uses a categorization task between a target concept and attribute. The WEAT uses a similarity measure to test for bias in the corpus. We investigate whether the same biases are present when using a representation for words that allows us to model categorization.

2 Method

In order to use this measure of graded hyponymy we must represent the meaning of words as a collection of their hyponyms. Here we used two different sources of hyponymy: WordNet [3] and Microsoft's Concept Graph [8]. We can then construct the representation of a word by adding together all the positive operators

of a specific word. We use positive operators because they have an ordering to them called the Löwner ordering which can be interpreted as categorization. To build the representations of single vector positive operators we take the outer product of the vector representation of a word with itself, more specifically we use use GloVe embeddings [9] as the source of word vectors in line with Caliskan et. al. [2]. By using this representation we can now define graded categorization using two methods described in Lewis [6] in terms of graded hyponymy (K_E and K_{BA}). The WEAT tests for implicit biases by comparing the similarity of two sets of comparable target words (*e.g., female names vs. male names*) against two sets of opposing attribute words (*e.g., pleasant attributes vs. unpleasant attributes*) with a null hypotheses that states that there is no difference between the similarity of either set of target words with the target concepts. We followed Caliskan et. al. [2] by using the same method but instead of measuring the association in terms of similarity, we measured the association in terms of categorization. The resulting test method for implicit biases in corpora is called the Positive Operator Association Test (POAT).

3 Results

Table 1 shows that the POAT is able to replicate most of the same results of the WEAT. The POAT performs well on non-offensive experiments such as the differential association between *flowers vs. insects* and *pleasant vs. unpleasant*. Additionally, the POAT records stronger stereotyping in the association tested in the last two rows, as well as in the *European-American vs. African-American – pleasant vs. unpleasant* experiment (row 5, Table 1) when using the attributes from the *young vs. old people's names* experiment. Two experiments that were not replicated well by the POAT are those in row 7 and 8. The first one instead shows a reversed association, due to inconsistent hyponymy representation, and the second records a low effect size and a high likelihood of the null hypotheses holding up.

Some target words had very low numbers of hyponyms, which skewed the results. To alleviate this problem, which was the case for most experiments that performed poorly on the POAT in Table 1, we use the same measure as in the regular POAT, but build the positive operators from the single word embeddings for each specific word. The results for these tests are presented in Table 2 and the largest difference in the rows 7 and 8: both now show large positive effect sizes and slightly smaller p-values compared to those of the WEAT. This version of the POAT performs best on all tested IAT findings. The found effect sizes are closer to the IAT effect sizes in 6 out of 8 experiments compared to the WEAT.

Discussion and outlook

In nine out of ten experiments the POAT is able to correctly recognize implicit biases in the word embeddings. Although the POAT was able to recognize the implicit biases, the strength that it recorded was sometimes not comparable to

	Target words	Attribute words			IAT		WEAT		POAT	
			N_T	N_A	d	P	d	P	d	P
1	Flowers vs. insects	Pleasant vs. unpleasant	25×2	25×2	1.35	10^{-8}	1.50	10^{-7}	**1.39**	10^{-6}
2	Musical instruments vs. weapons	Pleasant vs. unpleasant	25×2	25×2	1.66	10^{-10}	**1.53**	10^{-7}	1.47	10^{-7}
3	European-American vs. African-American	Pleasant vs. unpleasant	32×2	25×2	1.17	10^{-5}	**1.41**	10^{-8}	0.89	10^{-3}
4	European-American vs. African-American	Pleasant vs. unpleasant †	16×2	25×2	–	–	1.50	10^{-4}	1.04	10^{-2}
5	European-American vs. African-American	Pleasant vs. unpleasant ‡	16×2	8×2	–	–	1.28	10^{-3}	1.58	10^{-5}
6	Male vs. female names	Career vs. family	8×2	8×2	0.72	$< 10^{-2}$	1.81	10^{-3}	**1.68**	10^{-3}
7	Mental vs. physical disease	Temporary vs. permanent	6×2	7×2	1.01	10^{-2}	**1.38**	10^{-2}	−1.51	10^{-2}
8	Science vs. arts	Male vs. female	8×2	8×2	1.47	10^{-24}	**1.24**	10^{-2}	−0.001	0.50
9	Math vs. arts	Male vs. female	8×2	8×2	0.82	$< 10^{-2}$	**1.06**	10^{-1}	1.25	10^{-2}
10	Young vs. old people's names	Pleasant vs. unpleasant	8×2	8×2	1.42	$< 10^{-2}$	1.21	10^{-2}	**1.29**	10^{-2}

Table 1: Effect size (Cohen's d) and p-values for the WEAT and the POAT using the K_E measure and hyponyms derived from WordNet. Each row concerns a different implicit bias documented by the IAT. In each case the first (second) set of target words is found to be more compatible with the first (second) set of attributes words, N_T and N_A indicated the number of target words and attribute words, respectively. Bold values highlight the effect size closest to that of the IAT. † Attributes for this experiment are the same as in *Flowers vs. insects*. ‡Attributes for this experiment are the same as in *Young vs. old people's names*.

	Target words	Attribute words			IAT		WEAT		POAT	
			N_T	N_A	d	P	d	P	d	P
1	Flowers vs. insects	Pleasant vs. unpleasant	25×2	25×2	1.35	10^{-8}	1.50	10^{-7}	**1.30**	10^{-5}
2	Musical instruments vs. weapons	Pleasant vs. unpleasant	25×2	25×2	1.66	10^{-10}	**1.53**	10^{-7}	1.30	10^{-5}
3	European-American vs. African-American	Pleasant vs. unpleasant	32×2	25×2	1.17	10^{-5}	1.41	10^{-8}	**1.29**	10^{-6}
4	European-American vs. African-American	Pleasant vs. unpleasant †	16×2	25×2	–	–	1.50	10^{-4}	1.18	10^{-3}
5	European-American vs. African-American	Pleasant vs. unpleasant ‡	16×2	8×2	–	–	1.28	10^{-3}	1.46	10^{-4}
6	Male vs. female names	Career vs. family	8×2	8×2	0.72	$< 10^{-2}$	1.81	10^{-3}	**1.74**	10^{-3}
7	Mental vs. physical disease	Temporary vs. permanent	6×2	7×2	1.01	10^{-2}	1.38	10^{-2}	**1.26**	10^{-1}
8	Science vs. arts	Male vs. female	8×2	8×2	1.47	10^{-24}	**1.24**	10^{-2}	1.06	10^{-1}
9	Math vs. arts	Male vs. female	8×2	8×2	0.82	$< 10^{-2}$	1.06	10^{-1}	**1.00**	10^{-1}
10	Young vs. old people's names	Pleasant vs. unpleasant	8×2	8×2	1.42	$< 10^{-2}$	1.21	10^{-2}	**1.52**	10^{-2}

Table 2: This table shows the effect size (Cohen's d) and p-values for the WEAT and the POAT using the K_E measure, represented without hyponyms. All other settings are identical to those shown in Table 1

that of the WEAT or IAT due to inconsistent hyponymy representations of the target and concept words (row 7, Table 1). Neither sources of hyponymy we used contained an entry for every word. Nor do they contain all hyponyms of a word and in several cases the entry has zero hyponyms. Therefore, in order to make this method as dependable as possible the problematic word categories must be identified and remedied with some other method of deriving hyponyms. An example word type that for which this issue was prominent are the male and female pronouns that were part of the IAT experiments.

An advantage we found of our approach to detect biases in word meanings of the WEAT is that positive operators fit well inside a compositional framework [6]. This allows us to form phrases and sentences as well as generic sentences. Generic sentences such as "mosquitos carry malaria" express regularities. Using

positive operators gives the potential to assess associations between words and subphrases, such as mosquitos and carry malaria.

Our results shows that the use of the K_E measure as a proxy on single vector positive operators, where the words representations are not built using their hyponyms, outperforms the WEAT on six out of eight replications of IAT findings. This indicates that the use of an asymmetric measure to determine differential association is better at detecting implicit bias in word embeddings than the symmetric distance measure of the WEAT. The next step should be to identify exactly why the POAT performs so well on single vector positive operators.

References

1. Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior research methods **39**(3), 510–526 (2007)
2. Caliskan, A., Bryson, J., Narayanan, A.: Semantics derived automatically from language corpora contain human-like biases. Science **356**(6334), 183–186 (4 2017). https://doi.org/10.1126/science.aal4230
3. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
4. Greenwald, A.G., McGhee, D.E., Schwartz, J.L.: Measuring individual differences in implicit cognition: the implicit association test. Journal of personality and social psychology **74**(6), 1464 (1998)
5. Landauer, T.K., Dumais, S.T.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review **104**(2), 211 (1997)
6. Lewis, M.: Compositional hyponymy with positive operators. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). pp. 638–647. INCOMA Ltd., Varna, Bulgaria (Sep 2019). https://doi.org/10.26615/978-954-452-056-4-075, https://www.aclweb.org/anthology/R19-1075
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
8. Nsl, D.I.: Microsoft concept graph: Mining semantic concepts for short text understanding **1**, 262–294 (11 2019)
9. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1162, https://www.aclweb.org/anthology/D14-1162
10. Stubbs, M.: Text and corpus analysis: Computer-assisted studies of language and culture. Blackwell Oxford (1996)

Extended Abstract: Re-evaluating Knowledge Graph Embedding Models Performance on Domain Specific Datasets

Victor Ciupec[0000−0002−1025−9791] and Peter Bloem[0000−0002−0189−5817]

Vrije Universiteit, Amsterdam, The Netherlands
v.ciupec@student.vu.nl, p.bloem@vu.nl

Abstract. Knowledge graph embedding models (KGEs) have mostly been evaluated and compared on generic benchmark datasets. In this paper we research if training on domain specific datasets instead has any performance impact. We conducted an hyperparameter search experiment on five KGE models and found that the models perform generally better on domain specific datasets, although the relative performance and hyperparameter impact are in line with previous studies.

Keywords: knowledge graph embedding · domain specific dataset · hyperparameters · link prediction.

Introduction and Motivation. Knowledge graph embedding (KGE) models have become popular solutions for the link prediction problem in knowledge graphs. KGE models learn algebraic representations of the entities and relations in a knowledge graph and use a scoring function to predict and rank new triples, thus separating correct from incorrect triples. Various KGE models vary not only in their embeddings and scoring function, but also in the choice of hyperparameters, most notably loss function and training strategy. This paper researches the question on whether KGE models perform differently on domain specific datasets compared to generic ones like Freebase and Wordnet and specifically which of the studied models perform best and which hyperparameters impact performance the most. We investigate how certain properties of domain specific datasets such as ontological structure and redundancy in expressing facts influence the performance and the selection of hyperparameters. Our experiments follow the methodology in [3], with which we compare the performance metrics.

Experimental Setup. We trained and evaluated the KGE models in this study using two domain specific datasets AIFB [1] and MUTAG [2], both bound by strong ontological information, with very detailed schemas that contain full hierarchies of classes and sub-classes using the RDF model. We compare performance of five of the most popular KGE models: RESCAL, DistMult, TransE, ComplEx and ConvE. Our experiment uses quasi-random search across a large discrete hyperparameter space, followed by a Bayesian optimization for fine tuning numerical ones. The best model is selected using the entity ranking protocol metrics MRR and Hits@10 after training the five best configurations for each architecture.

		MRR	Hits@10
AIFB	RESCAL	42.1	56.9
	TransE	46.01	59.8
	DistMult	**49.2**	**60.1**
	ComplEx	48.7	60.0
	ConvE	47.2	58.7
MUTAG	RESCAL	35.63	46.65
	TransE	26.82	47.39
	DistMult	**48.07**	**60.32**
	ComplEx	38.68	50.49
	ConvE	31.63	47.39

Table 1. Performance on test data of the best performing models.

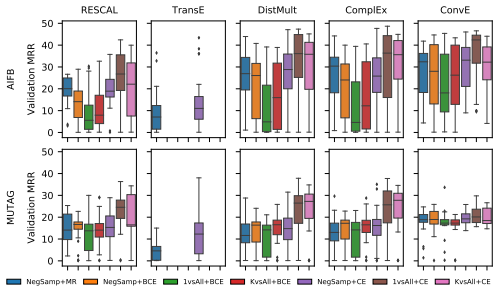


Fig. 1. Distribution of filtered MRR (%) on validation data over the train type/ loss function combinations during quasi-random hyperparameter search.

Results. In our study, DistMult has outperformed the other models on both domain specific datasets, followed by ComplEx. The relative performance gaps between the models is vastly reduced compared to the original publications, attributed perhaps to consistent training methodology. We also noted that the MRR and Hits@10 scores are remarkably higher for all models on AIFB than on MUTAG, justified by the existence of symmetrical relations in AIFB and by the KGE models' known ability to predict inverse relations.

We observed that the choice of loss function has by far the highest impact on the performance, followed by the training strategy. With some exceptions, cross entropy loss and 1vsAll training strategy performed best across the board. Furthermore, the higher MRR variance across the domain on AIFB suggests that models are more sensitive to hyperparameter change on AIFB than on MUTAG.

Compared with the results obtained by [3] on FB15K-237 and WNRR, most models performed notably better in our experiment, which can be explained by domain specific biases in our datasets. Our study[1] showed that domain specific datasets contribute to better KGE performance mostly due to the ontological structure and their intrinsic redundancy in expressing facts through the triples.

References

1. Bloehdorn, S., Sure, Y.: Kernel methods for mining instance data in ontologies. In: Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) The Semantic Web. pp. 58–71. Springer Berlin Heidelberg (2007)
2. Bühmann, L., Lehmann, J., Westphal, P., Bin, S.: Dl-learner structured machine learning on semantic web data. In: Companion Proceedings of the The Web Conference 2018. p. 467471. International WWW Conferences Steering Committee (2018)
3. Ruffinelli, D., Broscheit, S., Gemulla, R.: You CAN teach an old dog new tricks! on training knowledge graph embeddings. In: International Conference on Learning Representations (2020)

[1] https://github.com/Vixci/bachelor/blob/master/thesis.pdf

Applying Faster R-CNN and Mask R-CNN on the MinneApple Fruit Detection Challenge*

Callum Clark

Maastricht University, Paul-Henri Spaaklaan 1, 6229 EN Maastricht, Netherlands
c.clark@student.maastrichtuniversity.nl

1 Introduction

Technology has already had a major impact on the fruit industry. Image recognition for the detection and localisation of fruit is seen as a vital step to improve yield estimation[5] and therefore actual yields, as well as the automation of harvesting.

It is challenging to produce a generalised automation technique for all crop types because there are large differences in farming techniques across crop types. Previous research has been successful in developing specific techniques for individual crop types. In the case of apples, research has been done in autonomous apple picking[1][2], automated pruning[7] and yield estimation[8][5]. All of these tasks require the accurate detection and localization of apples. This is an important first step, since after detecting apples, one can estimate the size of the current harvest and make predictions on the final yield. Moreover, detecting and localizing apples can serve as a basis for assessing their health, detecting pests and thus support early intervention.

In this thesis we examine the problem of apple detection and localization as an Object Detection problem, applied to the challenging real-world dataset MinneApple[6].

We use the Facebook Artificial Intelligence Research's (FAIR) Detectron2 framework to train a Faster R-CNN and a Mask R-CNN on the MinneApple dataset and compare results to other state of the art methods.

Detectron2 provides a pretrained 101 layer ResNeXt network that we use as the backbone for both the Faster R-CNN and Mask R-CNN models. We then fine-tune these models by adjusting the learning rate and number of iterations to avoid overfitting.

2 Results

Our first set of results compares the AP scores of each model with various ResNet backbones. To identify the most suitable backbone network. Table 1 shows a

* This thesis was prepared in partial fulfilment of the requirements for the Degree of Bachelor of Science in Data Science and Knowledge Engineering, Maastricht University. Supervisor(s): Alexia Briassouli.

comparison of the AP@[.5:.05:.95] scores with different ResNet backbones. We see that networks with a ResNeXt-101 backbone score the highest. This is expected, as larger Residual Networks tend to be able to extract more salient features better than smaller ones.

Table 1. Comparison of different ResNet backbone networks

Method	Backbone	AP
Faster R-CNN	ResNet-50	0.398
	ResNet-101	0.425
	ResNeXt-101	0.436
Mask R-CNN	ResNet-50	0.386
	ResNet-101	0.394
	ResNeXt-101	**0.441**

Our next set of results presents our AP scores for all 6 COCO scoring categories, with AP@[.5:.05:.95] being our primary challenge metric. Table 2 compares our models' evaluation results to the benchmarked results, as well as the current second best score in the CodaLab competition. The competition server does not give information on what methods the other entries have used but we have included them for comparison.

2.1 Discussion

From our experiments, we can see that our proposed method has had success in beating other benchmarked scores. The use of a more accurate model can lead to a more accurate yield estimation with practical benefits. If scaled, this could have a major effect on reducing food waste in the agricultural production sector by increasing yield. As mentioned previously, this sector has the largest food waste in the food supply chain.

As expected, the deeper ResNet backbones provided a higher score. With the 101 layer ResNeXt network performing the best for both Faster R-CNN and Mask R-CNN.

The benchmark results provided by the challenge state that Faster R-CNN is the best performer. However, an interesting insight is that our Mask R-CNN outperforms our Faster R-CNN. This could be due to the Mask R-CNN using semantic segmentation and may have learnt to deal with clusters of apples better than the Faster R-CNN which only uses bounding boxes.

3 Conclusions

We can see that Detectron2's Mask R-CNN with a ResNeXt101 backbone achieves state of the art accuracy on the MinneApple Fruit Detection challenge. At time of publishing our technique sits at the first place on the challenge leaderboards.[3]

Table 2. Comparison with Benchmark

Author	**Method**	**AP@ IoU[.50:.05:0.95]**	**AP@ IoU=.50**	**AP@ IoU=.75**	**AP***small*	**AP***medium*	**AP***large*
Hani[6]	Faster R-CNN	0.438	0.775	**0.455**	0.297	0.578	**0.871**
Hani[6]	Mask R-CNN	0.433	0.763	0.449	0.295	0.571	0.809
Kuka[4]	NA	0.436	0.770	0.453	0.285	0.592	0.872
Ours	Faster R-CNN	0.436	0.791	0.436	0.291	0.590	0.848
Ours	Mask R-CNN	**0.441**	**0.801**	0.440	**0.300**	0.589	0.861

References

1. Baeten, J., Donné, K., Boedrij, S., Beckers, W., Claesen, E.: Autonomous fruit picking machine: A robotic apple harvester. In: Field and service robotics. pp. 531–539. Springer (2008)
2. De-An, Z., Jidong, L., Wei, J., Ying, Z., Yu, C.: Design and control of an apple harvesting robot. Biosystems engineering **110**(2), 112–122 (2011)
3. Haeni, N.: MinneApple Fruit Detection Challenge (2020 (accessed August 5, 2020)), https://competitions.codalab.org/competitions/21718
4. Haeni, N.: MinneApple Fruit Detection Challenge Results (2020 (accessed August 5, 2020)), https://competitions.codalab.org/competitions/21718results
5. Häni, N., Roy, P., Isler, V.: A comparative study of fruit detection and counting methods for yield mapping in apple orchards. Journal of Field Robotics **37**(2), 263–282 (2020)
6. Häni, N., Roy, P., Isler, V.: Minneapple: A benchmark dataset for apple detection and segmentation. IEEE Robotics and Automation Letters **5**(2), 852–858 (2020)
7. He, L., Schupp, J.: Sensing and automation in pruning of apple trees: A review. Agronomy **8**(10), 211 (2018)
8. Wang, Q., Nuske, S., Bergerman, M., Singh, S.: Automated crop yield estimation for apple orchards. In: Experimental robotics. pp. 745–758. Springer (2013)

Cooperation in Harsh Environments: The Effects of Noise in Iterated Prisoner's Dilemma

Louis Gevers[1] and Neil Yorke-Smith[1][0000−0002−1814−3515]

Delft University of Technology, The Netherlands
L.M.C.Gevers@student.tudelft.nl,n.yorke-smith@tudelft.nl

Abstract. Interactions in the real world are subject to mistakes and miscommunications. The presence of this noise challenges cooperation, as one party cannot determine whether the other party did not cooperate on purpose. Prisoner's dilemma strategies like Tit-for-Tat (TFT) perform badly once noise is present. Recent studies that harsh environments promote cooperation do not take noise into account. We show that the adversity of the environment benefits cooperators and can make cooperation more robust against mistakes. Harsher environments also encourage greater generosity to cope with noise. Yet when uncertainty is substantial due to higher probability of mistakes or more potential defectors in the environment, contrite behaviours are the most successful.

1 Motivation and Experimental Design

When environmental adversity is high, cooperation in many species counter-intuitively increases [1]. In spatial prisoner's dilemma (PD) games with simulated harshness, defecting strategies benefit the most at first, but in the long run cooperating strategies recover and later dominate the game [3]. Information in real-world interactions is not perfect, however, and errors occur. Strategies that perform well in a normal PD setting often fail when even occasional mistakes happen [5]. While various works have studied the effects of noise in spatial PD, it is unknown how these strategies fare in a harsh environment.

In order to simulate communication errors, we introduce an extra parameter for noise, E, into a spatial iterative PD simulation. E is the probability that the desired action of an agent actually results in the opposite action. Under different levels of cost-of-life, we study 12 different strategies: 9 'classical' strategies (ALLC, ALLD, RAND, GRIM, TFT, TFTT, STFT, TTFT, Pavlov) [2] and 3 strategies adapted to handle noise: *Generous Tit for Tat* (GTFT) and *Generous Pavlov* (GPavlov), and *Contrite Tit for Tat* (CTFT) [5].

2 Results and Discussion

The success of generosity in noisy environments has been widely supported [5]. We find that the importance of generosity is emphasized when the harshness of the environment increases. This contrasts with the reported success of less

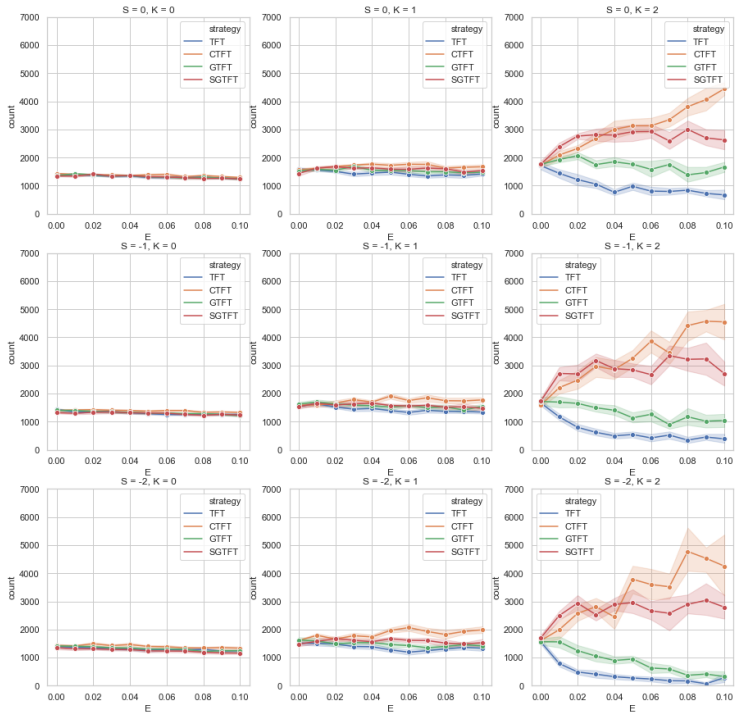


Fig. 1: Influence of noise in a tournament of TFT variants with varying harshness. S = sucker's payoff, K = cost of life, E = level of noise.

generous strategies such as GRIM in harsh environments [4], which confirms the importance of studying the PD with noise. Second, under higher noise, contrition is found to outperform generosity. The same phenomenon can be observed in the classic IPD [5]. The advantage of CTFT is that it works well in overtaking environments with defectors, while generous strategies rely on the presence of other cooperating strategies to succeed.

References

1. Andras, P., Lazarus, J., Roberts, G.: Environmental adversity and uncertainty favour cooperation. BMC Evolutionary Biology **7**(1), 240 (2007)
2. Jurišić, M., Kermek, D., Konecki, M.: A review of iterated prisoner's dilemma strategies. In: Proceedings of 35th Intl. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO'12). pp. 1093–1097 (2012)
3. Smaldino, P.E., Schank, J.C., McElreath, R.: Increased costs of cooperation help cooperators in the long run. The American Naturalist **181**(4), 451–463 (2013)
4. van Tilburg, J., Yorke-Smith, N.: Strategies for the iterated prisoner's dilemma in a natural environment. In: Presented at BNAIC'18 (2018)
5. Wu, J., Axelrod, R.: How to cope with noise in the iterated prisoner's dilemma. Journal of Conflict resolution **39**(1), 183–189 (1995)

Thesis abstract: Tensor-based pattern recognition, data analysis and learning*

S. Hendrikx[1,2], N. Vervliet[1,2], M. Boussé[1,2], and L. De Lathauwer[1,2]

[1] Department of Electrical Engineering (ESAT), KU Leuven, Belgium.
[2] Group Science, Engineering and Technology, KU Leuven Kulak, Belgium.

The aim of this thesis is to develop a scalable algorithm for multilinear regression [1]. Multilinear regression resides between linear and nonlinear regression models, such as neural networks (NN), which are widely used machine learning tools. While linear regression is simple and interpretable, it is less capable of modeling complex phenomena than its nonlinear counterpart. Multilinear regression is a trade-off between both, resulting in an expressive model with more meaningful variables than in NNs, as it is based on multilinear algebra. Unfortunately, the number of coefficients of a multivariate polynomial depends exponentially on its degree. In this thesis, we employ a low-rank tensor decomposition to break this exponential dependency, known as the curse of dimensionality (CoD), allowing us to develop a scalable, optimization-based algorithm.

Tensors, or multiway arrays, are higher-order generalizations of vectors and matrices. Tensor analogues of established matrix decompositions are powerful tools in signal processing, data analysis, and machine learning [2]. The main advantage of using tensor decompositions within this work stems from their ability to break the CoD. The polyadic decomposition (PD) decomposes an Nth-order tensor $\mathcal{T}$ as a sum of rank-1 tensors, where a rank-1 tensor is equal to the outer product, denoted by $\otimes$, of N nonzero vectors:

$$\mathcal{T} = \sum_{r=1}^{R} c_r \mathbf{b}_r^{(1)} \otimes \cdots \otimes \mathbf{b}_r^{(N)} \stackrel{\text{def}}{=} \left[\!\!\left[\mathbf{c}; \mathbf{B}^{(1)}, \cdots, \mathbf{B}^{(N)} \right]\!\!\right]. \tag{1}$$

An Nth-order tensor of size $I \times I \times \cdots \times I$ suffers from the CoD since it contains I^N entries. By approximating such a tensor with a low-rank PD, i.e., a PD with low R, that only has NRI parameters, this curse is broken.

An Nth-degree homogeneous polynomial $p(\mathbf{x})$ with variables $\mathbf{x} \in \mathbb{R}^I$ can be expressed by means of a symmetric[3] tensor of order N and the mode-n product[4]:

$$p(\mathbf{x}) = \mathcal{T} \cdot_1 \mathbf{x}^{\mathrm{T}} \cdot_2 \mathbf{x}^{\mathrm{T}} \cdots \cdot_N \mathbf{x}^{\mathrm{T}}. \tag{2}$$

* This research received funding from the Flemish Government (AI Research Program). This work was supported by the Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS Project no 30468160 (SeLMA). KU Leuven Internal Funds: C16/15/059 and IDN/19/014. Nico Vervliet is supported by a Junior postdoctoral fellowship (12ZM220N) from the Research Foundation—Flanders (FWO).

[3] A symmetric tensor is invariant to every possible permutation of its N dimensions.

[4] The mode-n product $\cdot_n$ of $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\mathbf{B} \in \mathbb{R}^{J \times I_n}$ is defined as $(\mathcal{A} \cdot_n \mathbf{B})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} a_{i1 \ldots i_N} b_{j i_n}$.

For example, a degree-two homogenous polynomial can be written as $p(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{T}\mathbf{x} = \mathbf{T} \cdot_1 \mathbf{x}^{\mathrm{T}} \cdot_2 \mathbf{x}^{\mathrm{T}}$. We assume that $\mathcal{T}$ has approximately low rank. This makes sense as real-life data can often be modelled using parsimonious representations due to some inherent structure. For example, measurements of a physical property that are governed by underlying differential equations or the dataset of a recommender system that contains users which behave in similar patterns. Compact models such as low-rank matrix and tensor decompositions are often used for large-scale problems in scientific computing and compressed sensing [3].

Thanks to this low-rank assumption, $\mathcal{T}$ in Eq. (2) can be replaced by a low-rank PD. The symmetry of $\mathcal{T}$ is exploited to obtain an even more compact model by using a symmetric PD, i.e., $\mathbf{B}^{(n)} = \mathbf{B}$ for $1 \leq n \leq N$, which requires fewer parameters than a general PD, namely RI. Therefore this model avoids the CoD since a multivariate polynomial generally has $\frac{(I+N-1)!}{N!(I-1)!}$ coefficients.

In order to derive an optimization-based algorithm, we fit the PD-constrained regression model to a dataset $\mathbf{X} \in \mathbb{R}^{I \times M}, \mathbf{y} \in \mathbb{R}^{M}$ as the following set of linear equations with a structured solution in a compressed-sensing style approach [4]:

$$\mathbf{y} \approx (\mathbf{X} \odot^{\mathrm{T}} \mathbf{X} \odot^{\mathrm{T}} \cdots \odot^{\mathrm{T}} \mathbf{X}) \operatorname{vec}(\llbracket \mathbf{c}; \mathbf{B}, \mathbf{B}, \cdots, \mathbf{B} \rrbracket), \tag{3}$$

where $\odot^{\mathrm{T}}$ denotes the row-wise Khatri–Rao product.

To compute the model variables $\mathbf{B}$ and $\mathbf{c}$, we use a Gauss–Newton (GN) algorithm with dogleg trust region to minimize the cost function $\frac{1}{2}||\mathbf{r}||_2^2$ in which $\mathbf{r}$ equals the difference between the left and right hand side of Eq. (3). By simultaneously exploiting both the Khatri–Rao and PD structure in Eq. (3) in the derivation of the cost function, gradient, Jacobian and Gramian for the GN algorithm, we obtain a scalable algorithm. Indeed, the overall per-iteration complexity of the algorithm is $\mathcal{O}(MR^2I^2)$, in contrast to $\mathcal{O}\left(MRNI^N\right)$ for a naive algorithm that does not exploit all structure.

To conclude, we have formulated a scalable optimization-based algorithm for multilinear regression through the use of a low-rank symmetric PD. In [5], we demonstrate high accuracy of our model on a materials science dataset.

References

1. Hendrikx, S. Tensorgebaseerde Patroonherkenning, Data-analyse En Machinaal Leren. Leuven: KU Leuven. Faculteit Ingenieurswetenschappen, 2019.
2. Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E. and Faloutsos, C. (2017). Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing, 65(13), 3551-3582.
3. Grasedyck L., Kressner D., Tobler C. A literature survey of low-rank tensor approximation techniques. GAMM-Mitteilungen. Feb 2013; 36(1):53–78.
4. Boussé, M., Vervliet, N., Domanov, I., Debals, O. and De Lathauwer, L. (2018). Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications. Numerical Linear Algebra with Applications, 25(6), e2190.
5. Hendrikx, S., Boussé, M., Vervliet, N., and De Lathauwer, L. (2019). Algebraic and optimization based algorithms for multivariate regression using symmetric tensor decomposition. In 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) (pp. 475-479).

Teaching a Machine to Diagnose a Heart Disease
Beginning from digitizing scanned ECGs to detecting the Brugada Syndrome (BrS) *

Simon Jaxy[1,2][0000−0002−7076−4108], Isel Grau[2][0000−0002−8035−2887], Nico Potyka[1][0000−0003−1749−5233], Gudrun Pappaert[3][0000−0001−9912−7121], Catharina Olsen[3][0000−0002−6193−2296], and Ann Nowé[2][0000−0001−6346−4564]

[1] University of Osnabrück, 49068 Lower Saxony, Germany
[2] Vrije Universiteit Brussels, 1050 Brussels, Belgium
[3] Universitair Ziekenhuis Brussel, 1090 Jette, Belgium

The Brugada Syndrome (BrS) is a severe cardiovascular disease that can lead to a sudden cardiac death even in patients with structurally normal hearts [2]. Ever since its first description [2], only one clear diagnostic case is acknowledged, characterized by an anomaly in the electrocardiogram (ECG); an accentuation of the J wave found in the right precordial leads (V1, V2), which results in an ST-segment elevation that is often followed by a negative T-wave [2], see Figure 1.

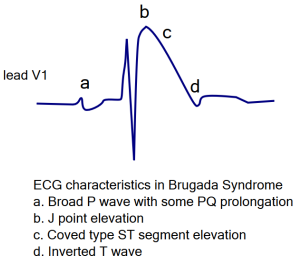


Fig. 1. The BrS positive pattern [7].

In the following, we present an automated pipeline that transformed scanned images of ECGs to time-voltage data, which is then used as basis for our long short-term memory (LSTM) [5, 3] classifier able to differentiate BrS positive ECGs from negative ones.

The digitization process follows an automatic pipeline that transforms scans of ECG images, comprising three distinctive image types (e.g., background and foreground color). A full description of the process can be found in [6].
First, the images are gray-scaled and rotated if needed. Then, obstacles, such as a black frame surrounding the signals or the background grid, are removed. Thirdly, we split the sequences into distinctive images by summing over the pixels in its columns. We use the minima in between peaks of pixels, each representing a single signal, as cut off points. Finally, every signal is upsampled and then mapped to time-voltage coordinates. For most of the ECG leads, the pipeline preserves the signals, see Figure 2. Yet, some sequences cannot be separated, leading to distortion.

The classifier's task is to read-in ECG images and make a binary decision of whether it is BrS positive or negative. We gathered positive ECGs (30 in total) with our pipeline while extracting negative examples (80 in total) from the PTB Database of Physionet [1], [4]. Our model is composed of a single LSTM-layer followed by a dropout layer and a sigmoid activation function for classification, its entire architecture and training process is described in the thesis [6].

* This work was supported by the project IRP8: IMAGica: an Integrative personalized Medical Approach for Genetic diseases, Inherited Cardiac Arrhythmias as a model.

Given the limited amount of data, the classifier scores a high amount of false positives, while avoiding any false negatives, see Table 1. In a medical context this might be favorable, as one would prefer to perform additional test rather than to miss a diagnosis, however, before it becomes relevant for any practical purpose, further improvements and testing has to be conducted.

		True		
		BrS+	BrS−	Total
Predicted	BrS+	22	45	67
	BrS−	0	24	24
	Total	22	69	91

Table 1. Confusion Matrix of our LSTM model.

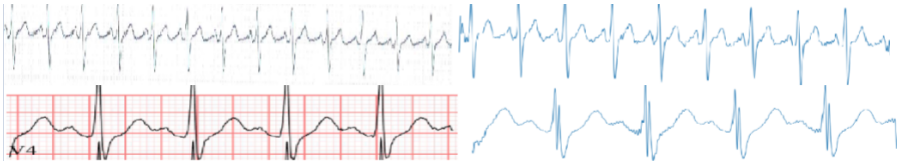


Fig. 2. The outcome of the digitization process for two ECG image types. The top image displays a good result the bottom image a distorted one.

We presented an automated pipeline capable of transforming scanned ECGs to time-voltage data. Furthermore, we explore the capabilities of the LSTM-based classifier on differentiating BrS positive ECGs from negative ones. Further research and experimentation are needed for obtaining a classifier achieving better performance. Another aspect is to investigate the role of different segments of the ECGs in the classification as positive for BrS.

References

1. Bousseljot, R., Kreiseler, D., Schnabel, A.: Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. Biomedical Engineering / Biomedizinische Technik **40**(s1), 317–318 (1995)
2. Brugada, P., Brugada, J.: Right bundle branch block, persistent st segment elevation and sudden cardiac death: A distinct clinical and electrocardiographic syndrome. Journal of the American College of Cardiology **20**(6), 1391–1396 (1992). https://doi.org/10.1016/0735-1097(92)90253-J, http://www.onlinejacc.org/content/20/6/1391
3. Gers, F., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. In: 9th International Conference on Artificial Neural Networks: ICANN '99. pp. 850–855. Institution of Engineering and Technology (1999)
4. Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. Circulation **101**(23), e215–e220 (2000)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
6. Jaxy, S.: Teaching a machine to diagnose a heart disease; beginning from digitizing scanned ecgs to detecting the brugada syndrome (brs) (2020)
7. de Jong, J.: Brugada ecg characteristics (2013), https://en.ecgpedia.org/images/f/f1/Brugada_ecg_characteristics.svg, last accessed 27 Aug 2020

Combining Structure from Motion with visual SLAM for the Katwijk Beach dataset

Marlon B. de Jong
Thesis supervisor: Arnoud Visser

Intelligent Robotics Lab, University of Amsterdam, The Netherlands

1 Introduction

In 2015 the Katwijk beach was used to imitate a Martian landscape to allow for a planetary rover to collect an abundance of data from multiple sensors [4]. This dataset is available for public use, with the intention of applying and testing Simultaneous Localization And Mapping (SLAM) techniques. For this thesis the focus was on the data collected from the stereo camera mounted on the rover. The research goal was to create an accurate 3D map with a correct location description. To reach this goal three SLAM techniques were applied to the stereo camera data. These techniques were: visual SLAM using Point clouds, Structure from Motion, and a combination of both.

2 Visual SLAM using Point Clouds

Visual SLAM was performed by first rectifying the left and right images of the stereo camera, followed by combining both images into a disparity map. The result is a point cloud, with for each point a distance estimate and a color. The point clouds are down-sampled into a 3D grid, with an average color for each cube. Following, the cubes are matched with the Iterative Closest Point algorithm [2]. By combining all point clouds one can create a 3D map and estimate the trajectory of the rover.

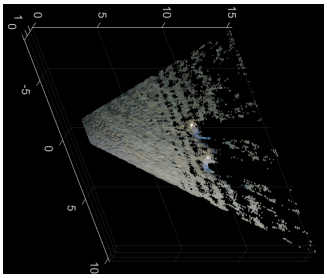


Fig. 1: Example of merged point clouds, depicting two stones in 3D.

The Visual SLAM maps were accurate when applied locally, but when applied on longer trajectories it is difficult to maintain a correct location estimation, because a majority of the points in the cloud display patterns which can be found on multiple locations.

3 Structure from Motion

Structure from Motion works differently in that it uses a limited number of key points in the landscape to map the environment, rather than the uniform distributed point clouds down-sampled in a grid. In this thesis Speeded-Up Robust Features (SURF) was chosen to identify the key points, because it is invariant to scale and photometric variations [1].

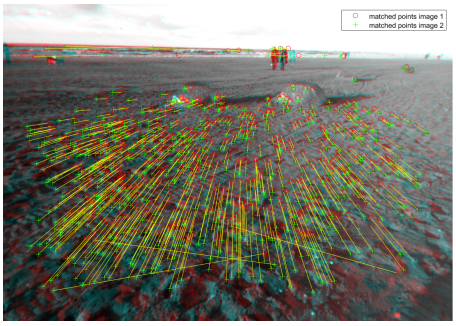


Fig. 2: Visualisation of the matching key points across two images.

The result is a trajectory estimate more accurate than the visual SLAM estimate, however the 3D map was visually incomprehensible from a human standpoint. This is because the map consisted of colourless points. Furthermore are the features close by more prominently represented, the stones were in most instances not close enough to register, as can be seen in Figure 2.

4 Combination

Both techniques have aspects which could be improved. The camera position estimate of visual SLAM is vulnerable to accumulation errors. Structure from Motion suffers from difficult map comprehension. A possible solution to both these problems would be to use the map representation from visual SLAM and the camera position estimate from Structure from Motion, to get the best of both approaches. The results of this combination can be found in the thesis [3].

5 Conclusion

Using the point clouds from visual SLAM and the location estimations from Structure from Motion, this thesis was able to create a visually more comprehensible map with a more accurate location estimation of the rover. The results indicate that by combining the techniques a better performance on both mapping and localisation can be achieved, therefore one would encourage further testing on the combination of these techniques.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision. p. 407. Springer (2006)
2. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image and Vision Computing **10**(3), 145–155 (1992)
3. de Jong, M.B.: SLAM at the Katwijk beach. Bachelor thesis, Universiteit van Amsterdam (July 2020)
4. Hewitt, R.A., Boukas, E., Azkarate, M., Pagnamenta, M., Marshall, J.A., Gasteratos, A., Visentin, G.: The Katwijk beach planetary rover dataset. The International Journal of Robotics Research **37**(1), 3–12 (2018)

Exploring the effects of conditioning Independent Q-Learners on the Sufficient Statistic for Dec-POMDPs

Alex Mandersloot[0000−0003−1617−2934], Frans Oliehoek[1][0000−0003−4372−5055], and Aleksander Czechowski[1][0000−0002−6054−9842]

[1] Department of Intelligent Systems, Delft University of Technology, Delft, The Netherlands

Abstract. In this study, we investigate the effects of conditioning Independent Q-Learners (IQL) not solely on the individual action-observation history, but additionally on the sufficient plan-time statistic for Decentralized Partially Observable Markov Decision Processes. In doing so, we attempt to address a key shortcoming of IQL, namely that it is likely to converge to a Nash Equilibrium that can be arbitrarily poor. We identify a novel exploration strategy for IQL when it conditions on the sufficient statistic, and furthermore show that sub-optimal equilibria can be escaped consistently by sequencing the decision-making during learning. The practical limitation is the exponential complexity of both the sufficient statistic and the decision rules.

Keywords: Deep Reinforcement Learning · Multi-Agent · Partial Observability · Decentralized Execution.

Introduction: The Decentralized Partially Observable Markov Decision Process (Dec-POMDP) is a widely used framework to formally model scenarios in which multiple agents must collaborate using private information. A key difficulty of a Dec-POMDP is that to coordinate successfully, an agent should decide on actions not only using its own action-observation history, but also by reasoning about the information that might be available to the other agents.

Independent Q-Learning (IQL) [1] is an easily-scalable multi-agent Reinforcement Learning method in which each agent concurrently learns the value of individual actions based on its individual information. It is well understood that such individual action-values are insufficient to capture the inter-agent dependency, and consequently IQL is not guaranteed to converge to the optimal joint policy. Instead, it is likely to converge to a joint policy that is in Nash Equilibrium [2]. However, such equilibria can be arbitrarily poor.

Precisely the obliviousness of IQL to the presence of other learning agents is our motivation for additionally conditioning IQL on the sufficient statistic for Dec-POMDPs [3], which contains a distribution over the joint action-observation history induced by the joint policy followed thus far. As a result, each agent is then equipped with an accurate belief over the local information available to the other agents, and is able to adjust its own behavior accordingly.

Experiments: We train a Deep Q-Network for each agent that conditions on the individual action-observation history $\overline{\theta}_t^i$ and the sufficient statistic σ_t, and learns the value of individual actions $Q_t^i(\overline{\theta}_t^i, \sigma_t, a_t^i)$. Methods are evaluated in the two agent Decentralized Tiger environment, whereby a horizon of 3 is employed.

To escape poor equilibria, an exploratory action of one agent should be observable to the others. To accomplish this, our agents explore in the space of entire *decision rules*. The sufficient statistic captures such decision rules, and thus facilitates the communication of exploratory actions among the agents. Importantly, however, the sufficient statistic summarizes only the *history* of joint decision rules. For *current* exploratory decision rules to be observable to others, we therefore additionally sequence the decision-making during learning. Specifically, agent 1 acts first and agent n is last to act. Each agent i then additionally conditions on the current (possibly exploratory) decision rules $\delta_t^{1:i-1}$ of the agents that acted before it to learn $Q_t^i(\overline{\theta}_t^i, \sigma_t, \delta_t^{1:i-1}, a_t^i)$. Our learners are able to consistently escape sub-optimal equilibria and learn the optimal policy, even when we explicitly force such equilibria upon the agents initially (Fig. 1).

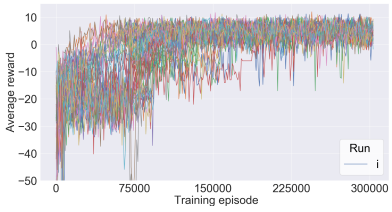


Fig. 1: All 50 learning curves.

Average Reward (std)	5.00 (0.77)
Ratio Optimal Policies	0.92

Table 1: Results across the 50 runs.

This project had received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

References

1. Tan, M.. Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the tenth international conference on machine learning. 1993. p. 330-337.
2. Boutilier, C.. Sequential optimality and coordination in multiagent systems. In: IJCAI. 1999. p. 478-485.
3. Oliehoek, F. A.. Sufficient plan-time statistics for decentralized POMDPs. In: Twenty-Third International Joint Conference on Artificial Intelligence. 2013.

Tracking Dataset use across Conference Papers

Pim Meerdink[12]
Supervised by Maarten Marx[1,3]

[1] University of Amsterdam
[2] pim.meerdink@student.uva.nl
[3] maartenmarx@uva.nl

Keywords: Named Entity Recognition · Coreference Resolution · Conference Papers

The enormous growth in the amount of papers published that the scientific community has experienced demands preliminary information extraction from scientific articles; due to time constraints researchers cannot read and understand all published scientific articles within their domain. Constructing knowledge bases containing information corresponding to these published articles has become an important task in streamlining scientific research. We work towards an end to end system that, given some large corpus of scientific articles, builds a bipartite graph containing dataset nodes on one side, and articles on the other. Dataset X and article Y will have an edge between them if and only if dataset X was used in article Y. This knowledge graph can be applied to, for example, extending a scientific literature search engine with a feature that allows users to explore datasets. The full paper can be found at [3].

The task at hand can essentially be divided into two sub-tasks. First there is **dataset mention extraction**, this entails identifying the phrases in the text that refer to a dataset. This task is an example of Named Entity Recognition (NER). Second, **entity clustering**, this entails partitioning the identified dataset mentions so that partition contains all the dataset mentions corresponding to one real world dataset. This task is an example of cross-document coreference resolution.

Allenai's sciBERT was used for the named entity recognition task, sciBERT is a BERT model pre-trained on scientific text [1]. A dataset of sentences containing dataset mentions was constructed using 15 000 scientific articles taken from NIPS, SIGIR, VISION and SDM. The final dataset contained 6000 BIO-labeled sentences, 2864 of these sentences had a dataset mention. The model was evaluated on a zero-shot test set, this entails that all of the datasets in this set (e.g. CIFAR-10) are not in the training data. The network obtained a tight fit of the training data, with an 'exact' f1 of 0.93. This fit reflected well onto both the evaluation and test set, where the F1 scores are 0.88 and 0.84, respectively.

When observing the performance of our model it becomes apparent that the model has little added difficulty correctly classifying dataset mentions that occur in sentences with more than 4 positively labelled instances. This means that the network is also able to understand and interpret ellipses and summations, these more complex rules and structures are not harder for the network to identify than

simple, one- or two-word dataset mentions. These structures and patterns are difficult even for human annotators to consistently parse and classify correctly, making the networks ability to understand the nuances of the labelling task significant.

For the entity clustering a task-specific algorithm was developed based loosely on [2]. The choice to divert from established practice and implement a custom solution was made in large part due to the specific nature of the entities to be clustered (i.e. they all describe datasets). This aspect of the problem allowed for assumptions and steps that improve performance significantly. Examples are assumptions that can be made with respect to the lexical structure of the entities, in particular the important role that numbers play in denoting datasets (CIFAR10 vs CIFAR100). In short, the developed algorithm first normalises the entities within the input data, and performs intra-document clustering: entities within each document are clustered using lexical similarity. Afterwards, a linear interpolation of similarities in a lexical, semantic and document level space are used to construct a graph G where each node represents the groups of dataset mentions found within an article. Lexical similarity was expressed in character level n-gram tf-idf cosine similarity, semantic similarity using sciBERT sentence embeddings cosine similarity and the document similarity was expressed using gensims doc2vec model trained on our corpus of 15 000 scientific articles. The edges in G express similarity, and all edges below a certain value are dropped. Each component in G now corresponds to an equivalence class of intra document coreferring entities.

The algorithm attained a B-cubed F1 score of 0.86. When performing grid search of the linear interpolation parameters of the lexical, semantic and document similarities it was found that the algorithm relied heavily on the lexical distance, while also using document level information. The sciBERT sentence embeddings expressing semantic similarity did not add much to the models ability to correctly cluster dataset mentions, and the top performing set of parameters did not use them at all.

Several steps must be taken before the developed system is ready to be deployed and utilized in a practical setting. First, the entity clustering algorithm must be expanded to parse ellipses and summations separately, and split them into their separate elements. Further, the computational complexity of the entity clustering remains an issue, due to the distance based nature of the algorithm it complexity scales quadratically with the input size. Finally, end-to-end evaluation should be performed of the system. While the systems' performance for each of the two subtasks was thoroughly evaluated, the overall, end to end system was not evaluated properly. This is, of course, an essential step in the development and deployment of the system.

References

1. Beltagy, I., Cohan, A., Lo, K.: Scibert: Pretrained contextualized embeddings for scientific text. CoRR **abs/1903.10676** (2019), http://arxiv.org/abs/1903.10676

2. Dutta, S., Weikum, G.: Cross-document co-reference resolution using sample-based clustering with knowledge enrichment. TACL **3**, 15–28 (12 2015). https://doi.org/10.1162/tacl_a_00119
3. Meerdink, P.: Tracking dataset use across conference papers (2020), `https://scripties.uba.uva.nl/search?id=715259`

Unsupervised clustering of groups with different selective attentional instructions using physiological synchrony

Alexandre Merasli[1], Ivo V Stuldreher[1,2][0000-0002-6104-2162], Anne-Marie Brouwer[1][0000-0003-1961-4291]

[1] Netherlands Organisation for Applied Scientific Research, Soesterberg, The Netherlands
[2] University of Twente, Enschede, The Netherlands
alexandre.merasli@outlook.fr, ivo.stuldreher@tno.nl, anne-marie.brouwer@tno.nl

Concurrent changes in physiological signals between people (Physiological Synchrony) can provide insight into attentional processes in groups, and individuals in relation to the group, for instance in educational settings. Earlier work showed that individuals can be correctly classified into attentional groups (attending or not attending to a story) based on the degree of synchrony with members of known groups (either instructed to attend or not attend to a story). Here we examine whether it is possible to find attentional groups from synchrony data using unsupervised learning, which may enable the identification of attentional groups without knowing anything about possible attentional foci beforehand as may be the case in real-world situations.

This study is based on data from [1] (publicly available on https://github.com/ivostuldreher/physiological-synchrony-selective-attention), where half of the participants were asked to focus on the story told by an audiobook, while the others were asked to focus on separate sound stimuli, occurring during the story. All participants heard the exact same audio file - only the instruction differed for the two attentional groups - and physiological signals were recorded during the experiment from three modalities (EEG - electroencephalography, EDA - electrodermal activity or skin conductance, and heart rate). From these signals, physiological synchrony was computed for each modality to assess the level of correlation between participants. Each synchrony coefficient represents the extent of synchronous change of physiological signals between two participants [1].

First, we investigated whether it is possible to classify people according to their attentional focus, by clustering synchrony values between participants as input. Clustering is a kind of method to find groups in the data without knowing any data labels (in this case, membership of the true attentional group). A straightforward approach is to directly apply specific clustering algorithms on the pre-processed synchrony data. Such kind of algorithms like hierarchical clustering or K-Medoids can cluster data without using coordinates but rather distances between points, which is the kind of information in the pre-processed synchrony data. The data appeared to be not easily clusterable, resulting in low performance (e.g. an accuracy of 65% when clustering EEG with hierarchical clustering). We then investigated adding a step before clustering, with different mapping methods which can be used to visualize and remove noise from the original synchrony data. Principal Coordinate Analysis (PCoA), Multi-Dimensional Scaling (MDS), or Uniform Manifold Approximation and Projection (UMAP) were used, with

the aim of finding coordinates which approximate the pre-processed synchrony data. Then, several clustering algorithms like K-Means, spectral clustering or hierarchical clustering were applied on these computed coordinates to provide two groups of participants. We studied each possible combination of mapping and clustering algorithms. We achieved the highest performance with EEG, with a classification accuracy of 85% obtained when mapping with PCoA and using spectral clustering. EDA and heart rate did not perform above chance level.

After demonstrating that unsupervised detection of attentional groups is possible for EEG, we continued to look into combining the information coming from different modalities (EEG, EDA and heart rate) in order to possibly enhance the information coming from only one modality. Mapping methods like Multi View Multi-Dimensional Scaling (MVMDS) or Multi View Spectral Clustering (MVSC), which can be seen as an extension of respectively PCoA and spectral clustering, enabled us to combine the information present in several input matrices to create a map with all participants. We found that adding EDA to EEG did not improve the accuracy of the results, but made them more robust to varying pipelines than using EEG alone: the worst result using EEG alone was 54% accurate, whereas the worst result when EEG and EDA were combined was 73% accurate. From all combinations and pipelines, the best result was achieved by combining EEG and heart rate, where results were more robust and more accurate than EEG alone, with an accuracy of 92% when applying K-Means after MVMDS. Combining three modalities rather than only adding EDA or heart rate to EEG did not seem to further improve performance.

Finally, we investigated how to approach the problem of choosing the proper classification pipeline for real world cases that our data may not generalize to. In the current study, we could use our known attentional groups to evaluate the clustering performance. However, in real-world applications, this ground truth information is usually not available. We therefore studied the silhouette coefficient to assess clustering quality, and investigated its correspondence with the ground truth accuracy. Unfortunately, the data appear to be too noisy to successfully use this coefficient to choose good clustering pipelines among all methods.

In sum, our study indicates that it is possible to use unsupervised clustering on physiological synchrony data to identify groups with different attentional foci. Performance is close to that reached using knowledge of attentional groups, i.e., classifying an individual into a known attentional group that he or she correlates with most strongly [1]. Similar as in [1], we found that physiological synchrony in EEG is more informative than EDA and heart rate. However, adding heart rate or EDA to EEG results in classification performance that depends less on the specific pipeline.

References

1. Stuldreher, I. V., Thammasan, N., van Erp, J. B., & Brouwer, A. M. (2020). Physiological synchrony in EEG, electrodermal activity and heart rate reflects shared selective auditory attention. Journal of Neural Engineering, 17(4), 046028.

The Maintenance of Conceptual Spaces Through Social Interactions

Max Peeperkorn[1], Oliver Bown[2], and Rob Saunders[1]

[1] LIACS, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
[2] University of New South Wales, Sydney, Australia
post@maxpeeperkorn.nl

For this research, a computational social creativity [6] simulation has been developed using the Variational Autoencoder (VAE) [4] as a computational model of conceptual spaces. Due to their probabilistic nature and their compression and generative capabilities, VAEs are a good fit for mechanising conceptual spaces. Based on these characteristics this research assumes that the VAE is a reasonable abstraction of conceptual spaces. Subsequently, the simulation investigates how the conceptual spaces are influenced by social interactions.

Creativity is a social phenomenon. As individuals share their perspectives, ideas emerge that none could have had on their own [7]. These perspectives are embedded in an individual's conceptual space, which plays a central role in the search for novel ideas and artefacts. In a general cognitive view, Gärdenfors [3] proposes conceptual spaces are geometric mental structures that organise thought. In the context of creativity, conceptual spaces are a key aspect in Boden's framework for creativity to support the explorative and transformative modes of creativity [1]. However, due to her abstract definition, it is difficult to use for computational purposes. By combining Boden's approach for examining and Gärdenfors' geometric view for traversing conceptual spaces, this research proposes to model conceptual spaces using VAEs.

The widely accepted systems view of creativity [2] is used as the basis of the simulation. In this view, creativity is observed in the interactions between its three components: *the domain,* the *individual,* and *the field*. The domain is an abstract cultural repository, the individual produces variations based on knowledge held in the domain, and the field is a social space for the individuals where variations are selected to be preserved in the domain.

By using VAEs to model each individual's conceptual space, the domain is distributed amongst all of the individuals. To allow the analysis of the simulation a pretrained global VAE is introduced as an overarching view of the distributed domain and determines the initial training of the agents. During the simulation, a recommender system uses a shortcut to the global VAE to serve as a matchmaker to find 'like-minded' agents, acting as a proxy for socio-cultural gatekeepers (e.g. art galleries) by enforcing the fields ideology.

Each round in the simulation consists of three steps. During the first step, each agent receives artefacts selected by the field, then trains their conceptual space and produces new variations guided by their preference for novelty. In the second step, the recommender system determines the position of each agent.

Finally, the artefacts for the next round are selected from the pool of artefacts produced by the agent and its neighbours, as determined by the fields' ideology.

The evaluation of the VAEs shows that finding the correct scale of artefacts selected and produced and the amount of training each round is crucial for the maintenance of conceptual spaces. Initially, a small scale and a high number of training epochs resulted in heavy overfitting from round to round. With an increased scale and a lower number of training epochs, the VAEs stabilized, suggesting the adaption to the artefacts selected by the field.

To investigate the influence of ideologies on the development of conceptual spaces distributed across fields, three were simulated: neutral, progressive and conservative. The neutral ideology is the same as no ideology and uniformly selects artefacts. The progressive ideology favours artefacts in less explored areas. The conservative ideology favours frequently and recently selected artefacts. As the progressive ideology pushes for exploring novelty it leads to a better maintenance of the conceptual spaces, exposing the individuals to a diverse range of artefacts. Conversely, the conservative ideology deteriorates, likely because the same artefacts are too frequently selected.

On the individual level, a novelty preference was introduced by changing the standard deviation used when sampling the latent spaces to generating new artefacts. Novelty can be viewed as the disruption of expectedness [5]. Sampling with a lower standard deviation produces less unexpected results, while a higher standard deviation pushes towards the edges where less information is embedded. The results show that a higher novelty preference leads to more distributed interaction between the agents (Fig. 1). While the maintenance is stable, the VAEs are less performant compared to simulations with lower novelty preferences.

The utility of Variational Autoencoders is demonstrated for use during the simulation and as a tool for analysing creative behaviours and output. Additionally, the influence of different ideologies of the field and novelty preferences of the individuals has been explored and shows that the maintenance of the VAE is aligned with the social interactions of the individuals. This contribution suggests the possibility of evaluating creative output on learned relations without the use of predetermined rules.

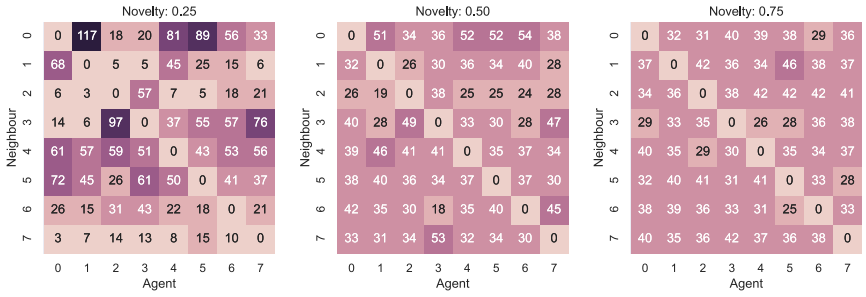


Fig. 1. The communication matrices indicate the number of interactions between the agents. With a low novelty preference, the individuals tend to stick with individuals with similar conceptual spaces, while higher preferences leads to more distributed communication.

References

1. Boden, M.: The Creative Mind: Myths and Mechanisms. Routledge, London (2004)
2. Csikszentmihalyi, M.: Society, Culture, and Person: A Systems View of Creativity, pp. 47–61. Springer Netherlands, Dordrecht (2014)
3. Gärdenfors, P.: Conceptual spaces: The geometry of thought. MIT press (2004)
4. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2013)
5. Martindale, C.: The clockwork muse: The predictability of artistic change. Basic Books (1990)
6. Saunders, R., Bown, O.: Computational social creativity. Artificial Life **21**(3), 366–378 (2015)
7. Vygotsky, L.S.: Imagination and creativity in childhood. Journal of Russian & East European Psychology **42**(1), 7–97 (2004)

Sequence-to-Sequence Speech Recognition for Air Traffic Control Communication

Tijs Rozenbroek
Thesis supervisors: Dr F.A. Grootjen and Dr U. Güçlü

Radboud University, Nijmegen, the Netherlands
t.rozenbroek@student.ru.nl

Keywords: Automatic Speech Recognition · Air Traffic Control · Sequence-to-Sequence Models

1 Introduction

Air Traffic Control (ATC) is absolutely essential in aviation, and air traffic controllers have the highly taxing job of directing all air traffic within a specific region, preventing accidents, giving information to pilots and more. Since ATC is so important, all efforts must be made to ensure that ATC communication happens optimally and efficiently, without error. Any effort to assist the air traffic controllers or pilots in their communication is therefore warranted.

An example of a system that can assist air traffic controllers and pilots is a system which detects errors in communication, such as errors in repeating instructions and callsigns, known as readback errors. What follows is an example of how an undetected readback error can lead to dangerous situations.

On the 7th of March 2016, at EuroAirport Basel Mulhouse Freiburg, a serious incident took place due to a combination of factors [1]. A readback error by a pilot was not corrected by the air traffic controller, which led to two planes being on the same runway at the same time. The planes missed each other by a mere 115 meters, which is a very small distance in these kinds of situations.

A system that could automatically detect this type of error, and warn pilots or controllers, could improve overall aviation safety. To build such a system, or a related system, a sufficiently fast and reliable automatic speech recognition (ASR) system is required.

Currently, few ASR systems have been developed for ATC on the whole. According to Helmke et al. [3], efforts to bring ASR into the domain of ATC have been made as early as the 1990s. However, no instances of using sequence-to-sequence models for ATC have been found, which is the gap in the field that this work aims to fill.

2 Background

2.1 Air Traffic Control

There are, unfortunately, factors that make the ATC domain a difficult domain to implement ASR into. These specifics are, amongst others, high levels of noise, non-native speakers (accents), standardised special phraseology and perhaps more importantly, deviations from this standard phraseology. Additionally, ATC communication is rapid and thus lightweight and fast ASR solutions are required.

2.2 Sequence-to-Sequence Model

The sequence-to-sequence model architecture that was taken as the basis for the experiments in this work, is the recently published model architecture by Hannun et al. [2]. The authors show that the model architecture performed well when trained and tested on the LibriSpeech dataset [4], where it attained a word error rate (WER) of 15.64 without an external language model, 11.87 in combination with a 4-gram language model and 9.84 with a convolutional language model, when evaluated on the slightly more challenging 'test-other' test set from LibriSpeech [2].

The model architecture's most interesting and novel feature is the use of time-depth separable (TDS) convolutions, which, as the authors claim, generalise better than other deep convolutional architectures and use fewer parameters.

3 Methods, Results and Conclusion

As mentioned, the model architecture by Hannun et al. was taken as the basis for the experiments in this work. Several approaches were taken for attempted improvements of the model architecture, ranging from increasing receptive fields of the aforementioned TDS convolutions, to increasing the amount of TDS layers. The training configuration was manipulated in several ways to improve convergence and thus improve performance.

The best-performing model that was made in this work, scored a word error rate of 26.19% on noisy, low-quality ATC data, and 5.9% on relatively clean data. It is important to mention that these tests were conducted without external language models, leaving room for further improvements. The high WER on the noisy data can be largely attributed to its noise, which caused some utterances to be nearly unintelligible, even to a trained ear. Addressing these issues would be key for improving performance, which could perhaps partially be done by improving the robustness of the model.

With these results in mind, it can be stated that in the future, sequence-to-sequence models in general might be a viable option for an ASR model for ATC, and time spent further developing these models would be well spent. All in all, a solid contribution to the field of automatic speech recognition for air traffic control has been made, since the absence of sequence-to-sequence models in this field has been concluded.

References

1. Serious incident to a dornier 328 registered HB-AEO and to an embraer 190 registered PH-EXB occured on 07/03/2016 at bâle-mulhouse (68) (2018), https://www.bea.aero/uploads/tx_elydbrapports/BEA2016-0122.en.pdf
2. Hannun, A., Lee, A., Xu, Q., Collobert, R.: Sequence-to-sequence speech recognition with time-depth separable convolutions. In: Interspeech 2019. pp. 3785–3789. ISCA (2019). https://doi.org/10.21437/Interspeech.2019-2460
3. Helmke, H., Ehr, H., Kleinert, M., Faubel, F., Klakow, D.: Increased acceptance of controller assistance by automatic speech recognition. In: Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013). pp. 1–10 (2013), https://elib.dlr.de/87600/
4. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: LibriSpeech: An ASR corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5206–5210. IEEE (2015). https://doi.org/10.1109/ICASSP.2015.7178964

Large Cone Beam CT SCan Image Quality Improvement Using a Deep Learning U-Net Model

Joel Ruhe[1], Valeriu Codreanu[2], and Pascal Wiggers[1]

[1] Amsterdam University of Applied Sciences
[2] SURFsara, Amsterdam

Abstract. Cone beam CT scanners use much less radiation than to normal CT scans. However, compared to normal CT scans the images are noisy, showing several artifacts. The UNet Convolutional Neural Network may provide a way to reconstruct the a CT image from cone beam scans.

1 Introduction

Many people die annually from the effects of cancer. Most common and one of the deadliest type of cancers is lung cancer [1]. Primarily, lung cancer is being monitored using a CT scanner. The regular dose of a lung cancer ct scan is 1.5 millisieverts. This amount of millisievers forces the patient to take long brakes between the CT scans [2]. Cone beam CT scans are CT scans with much less radiation that can achieve relatively high image quality. However, there is more noise in the image than a regular CT scan [3].

CT image quality depends on the following factors: image contrast, spatial resolution and image noise. A Cone Beam Computed Tomography (CBCT) its ray source is cone-shaped and a two-dimensional detector is used that makes one rotation so that a 'volume' of data is obtained. The main advantage of the cone beam CT scanner is that it uses much less radiation compared to normal CT scans. The main disadvantage is that because of this, the quality of the image goes down (black 'stains', or group of pixels, appear in the image, on top of the general CT image noise). Artificial neural networks may provide a solution to this problem. In particular, we developed a U-NET model to improve the cone beam CT images so that they are 'restored' as good as possible to the quality of a normal CT scan. The question we addressed in the research described in this paper is: *How can a U-NET model, consisting of convolutional and deconvolutional layers, be used to improve 3D cone beam image quality of lung CT scans?*

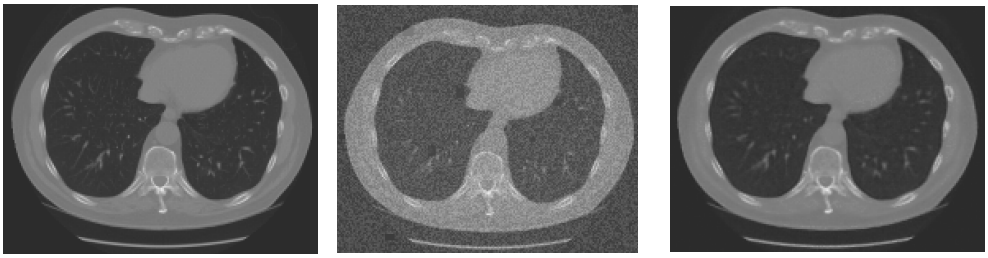
2 Model

Typical Convolutional Neural Networks (CNN's) consists of a convolution operation, Non-linearity (activation) function, pooling and fully connected layers. The convolutions create feature maps from the original input image. Because of this it can find recognizable patterns in the image. Different types of filters can be used depending on the type of feature maps one wants.

The U-NET model [4], developed to tackle the noise problem that occurs in CBCT images, consists of convolutional layers and transpose convolutional layers. This way, patterns can be found on 'what' is in the image, but also 'where' it is. When it is in the decoding path, it performs concatenation operations with the encoding blocks. This way, high resolution feature maps from the encoding blocks are being concatenated with the upsampled features. This will better learn representations with following convolutions and is also the main contribution of a U-NET model.

3 Experiments

To prove the principle, in this research a DICOM data set has been used that first had to be decoded to only get the raw pixel data on which the model can learn. Horovod has been used for data parallelism during the training. Later on, this might be extended to model parallelism to train on even larger image sizes if necessary. The U-NET model has been trained on 32x32x8, 64x64x16, 128x128x32, 256x256x128 and 512x512x128 image sizes. The images below show the results for an original image of size 256x256x64.



(a) Original image. (b) Cone beam image. (c) Reconstructed image.

From the image results, it is clearly noticeable that the U-NET model does a pretty good job in recreating the image with added noise. Only the small white matter in the middle of the lungs it has trouble with. We also found that the loss decreases as the image size increases. This also means that the U-NET model has greater precision on the 512x512x128 image due to the amount of feature maps it creates with each convolutional layer. The downside to this, is that it leaves a large memory footprint on the GPU when using large image size.

4 Conclusion

In short, it can be concluded that the U-NET model is suitable for recreating cone beam images and performs best at the image size 256x256x64. Furthermore, the U-NET may be directly applied to the CBCT images acquired from a commercial CBCT scanner after decoding the images and can directly be applied to real world problems.

References

1. World Health Organization Cancer. World Health Organization, September 12, 2018.
2. radiologyinfo. Radiation Dose in X-Ray and CT Exams. radiologyinfo, March 20, 2019.
3. Lee W. Goldman. Principles of CT: Radiation Dose and Image Quality. Journal of Nuclear Medicine Technology, November 15, 2007.
4. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

Safe tests for 2 x 2 contingency tables and the Cochran-Mantel-Haenszel test*

Rosanne J. Turner[1,2**]
Thesis supervisor: Prof. Peter. D. Grünwald[1,3]

[1] Machine Learning Group, Centrum Wiskunde & Informatica, Netherlands
[2] Brain Center, University Medical Center Utrecht, Netherlands
[3] Mathematical Institute, Leiden University, Netherlands

Keywords: Online learning · Bayesian learning · Information theory.

Rationale In my thesis, I developed implementations of *safe statistics*: a new framework for collecting evidence for hypotheses, particularly suitable for online and sequential learning [1]. Currently, p-values and (frequentist) confidence intervals are the most widely-used methods for collecting evidence for hypotheses. However, with these methods, error bounds are only guaranteed if the number of samples for each experiment *and* the number of experiments are fixed in advance. This means these statistics should not be used in an online setting (a prototypical example is A/B testing); would one do this anyway, the probability of obtaining false "significant" results would approximate 1 as the number of data points collected grows. Since feasible, easily implementable methods that are robust under online use have not been available to the research community, classical methods have been used anyway, with many expensive false-positive findings as a consequence.

Similarly, standard statistics also do not provide guarantees in the common situation that experiments (e.g. randomised trials) are conducted sequentially, when the decision to start a new experiment is based on previous results [3]. It directly follows that meta-analysis results, and even combined evidence from multiple experiments performed within the same research group can be misleading. The safe statistics framework provides methods that *can* be used to analyse data in real-time, and to effortlessly combine statistics from sequential experiments.

Safe statistics Within the safe statistics framework, random variables called E-variables[4] are used to represent the *evidence* for a hypothesis in the data. By definition, an E-variable is a nonnegative random variable that has an expected value of at most 1 under the *null hypothesis* $\mathcal{H}_0$. The higher an E-value, the more evidence there is in the data in favour of the *alternative hypothesis* $\mathcal{H}_1$. From

* Two-page abstract of the master thesis written by Rosanne. J. Turner at Leiden University for the Master Statistical Science for the Life and Behavioural Sciences, defended September 23, 2019, see [4].

** Corresponding author: Rosanne J. Turner, rosanne@cwi.nl

[4] called S-variables in the previous versions of the framework and my master thesis

the definition of E-variables, it can straightforwardly be derived that when we use the rule that we reject $\mathcal{H}_0$ when the E-value exceeds $\frac{1}{\alpha}$ for some $\alpha \in [0, 1]$, we have a test where the probability of falsely rejecting the null is bounded by α. The definition also implies that all E-variables can be used in the sequential setting simply by multiplying them. It also turns out that a special subset of E-variables can be used in the online testing setting [1].

To optimise the amount of evidence collected, an information-theoretic criterion for *good* E-variables was defined: GROW, which stands for *Growth Rate Optimal in the Worst case* [1]. GROW E-variables tend to grow fastest for some alternative hypothesis $\mathcal{H}_{1,\delta} : \{P_{\theta_1} : \theta_1 \in \Theta_1(\delta)\}$ defined by a distance metric δ, even *in the worst case* scenario where data are generated by a distribution in $\mathcal{H}_{1,\delta}$ that yields little evidence. It turns out that these GROW E-variables have the form of *Bayes factors* and can be derived for any pair of hypotheses $\mathcal{H}_1$ and $\mathcal{H}_0$ [1], but the corresponding prior distributions are sometimes completely different from what Bayesian machine learners or statisticians would normally use.

Results and short discussion For this thesis, I developed GROW E-variables equivalent to two classical frequentist hypothesis tests: the two-by-two contingency table test and its stratified version, the Cochran-Mantel-Haenszel test. Two versions of the E-variable were developed. For the first version, $\mathcal{H}_{1,\delta}$ was defined with δ the Kullback-Leibler divergence. This E-variable could be useful when one wants to design a test optimised for distributions that would yield a certain minimal growth rate if they would generate the data. For the second version, $\mathcal{H}_{1,\delta}$ was defined with δ the absolute difference between the proportions. Such an E-variable is useful when one has more clear ideas about the applied goal of the experiment and wants to detect a *minimal* difference between two groups.

For the 'minimal absolute difference' version, the GROW E-variable was derived analytically. I showed that when using this E-variable in an online, real-time fashion, the expected sample size needed to achieve a desired power can be lower than when using its classical equivalent, Fisher's exact test. No analytic expression could be found for the Kullback-Leibler version: this GROW E-variable has to be found through numerical optimisation. Nevertheless, the Kullback-Leibler version could still be preferred in some cases: it was shown to gain higher power for certain data-generating distributions compared to the absolute difference E-variable.

Both E-variables were implemented in the Safestats R package, a collaborative project with other machine learning researchers from Amsterdam [2]. The work in this thesis gave rise to some interesting follow-up questions, such as the development of 'most powerful' GROW E-variables, safe confidence sequences for proportions, and applications of E-variables for healthcare research, and is continued in my current PhD project.

References

1. Grünwald, P., de Heide, R., Koolen, W.: Safe testing. arXiv preprint arXiv:1906.07801 (2019)
2. Ly, A., Turner, R.J.: Safestats: an R package for safe, anytime-valid inference. `https://github.com/AlexanderLyNL/safestats`, accessed: 2020-08-28
3. Ter Schure, J., Grünwald, P.: Accumulation bias in meta-analysis: the need to consider time in error control. F1000Research **8** (2019)
4. Turner, R.J.: Safe tests for 2x2 contingency tables and the Cochran-Mantel-Haenszel test. Master thesis. `https://www.universiteitleiden.nl/binaries/content/assets/science/mi/scripties/statscience/2019-2020/thesis_rjturner_for_publication.pdf` (2019), accessed: 2020-10-22

Understanding Happiness by Using a Crowd-sourced Database with Natural Language Processing

Yixia Wang Giacomo Spigler

Department of CSAI, Tilburg University, Tilburg, The Netherlands
y.wang_1@tilburguniversity.edu,
g.spigler@uvt.nl

Abstract. In this thesis [1], we conduct two classification tasks on the crowd-sourced database Happy DB, which consists of more than 100,000 descriptions of happy moments collected using Amazon's Mechanical Turk. We apply the state-of-art word embedding algorithm BERT to transform all happy moments to context-sensitive representations and then feed them to a one-layer LSTM to learn two critical concepts of happiness, *agency* and *sociality*. We found that the proposed setup improves performance compared to the previous works.

1 Introduction

Natural language processing has been used to decipher human language. Tasks in this turf such as machine translation, speech recognition, and product recommendation, have vastly improved over the last few years. At the core of these language processing technologies are language models that transform massive amounts of textual information into multi-dimensional vector representations of words or sequences, which are then used as input representations in complex artificial intelligence tasks. This thesis focuses on learning two concepts of happiness, *sociality* and *agency* via two classification tasks. 'Sociality' here refers to feeling happy in the presence of others vs alone, while 'agency' denotes whether the happy moment refers to the participant who reported it or to other people.

Previous work explored diverse methods involving supervised and semi-supervised learning. The former includes a Word Pair Convolutional Model based on the hypothesis that a small set of word pairs were vital for representing the nature of sociality/agency of these happy moments [11], and similar models based on CNNs [12, 2, 15] and RNN/LSTM/Bi-LSTM[10, 13]. The latter comprises learning settings incorporating autoencoders [1] or k-means clustering [14]. Various embedding algorithms were also employed in this task including word2vec [6], GloVe [7], ELMo [8] and word embeddings pre-trained on WikiText-103 corpus[14]. Among all these models, the Elmo-based LSTM proposed by UBC [10] holds the state-of-the-art for prediction on agency(85%) and sociality(92%).

[1] Full thesis: http://spigler.net/giacomo/files/yixia_wang_thesis_2020.pdf

2 Methods

Our proposed method combines LSTMs with a state-of-art word embedding algorithm, BERT. The LSTM used had 60 hidden units and a Tanh activation function, while the output layer. Training was performed using mini-batch gradient descent (size=32) and the Adam optimizer [5] (learning rate $\eta = 0.1$). Early stopping [9] is used in addition to dropout (dropout rate=0.2) to reduce overfitting.
Out of 24 released BERT models, we use the BERT-Base model(Uncased: 12-layer Transformer, 768 dimensions, 12-heads, 110M parameters)[3]. Max sequence length is set at 128. Padding and truncation are used to fix the length of each account of happy moments to 128 tokens. Therefore, for each input text, BERT outputs a tensor of shape (128, 768) with one vector per token. Out of 12 layers, we summed the last four layers as a pooling strategy to obtain a fixed representation for each happy moment description.

We also developed eight baseline models by applying traditional machine learning algorithms, including Support Vector Machine (SVM), Random Forest, Logistic Regression (Log Reg) and Naive Bayes. Each of them is implemented with two sets of word embedding algorithms: Bag of Words (BOW) and Bag of Words with a TF-IDF transformation (BOW tf-idf). As most of the previous works [4] reported their highest accuracy from an architecture equipped with the GloVe word embedding, a further baseline based on LSTM + GloVe is also used.

3 Results

The results of the baseline models show that a linear SVM model with BOW as word representation performs the best overall (sociality accuracy=90.49%, agency accuracy=78.34%), although the classification of agency was found to be marginally higher using Logistic Regression + BOW (accuracy=80.65%). The results of the main evaluation are shown in Table 1. The proposed solution was found to improve all metrices (accuracy, F1 score, and AUC) compared to both GloVe+LSTM and ELMo+LSTM [4].

	Sociality			**Agency**		
Models	**Accuracy**	**F1 Score**	**AUC**	**Accuracy**	**F1 Score**	**AUC**
ELMo + LSTM(publication)	92.00%	93.00%	None	85.00%	90.00%	None
GloVe + LSTM	90.14%	90.89%	95.70%	83.70%	88.55%	89.41%
BERT + LSTM	**93.00%**	**93.49%**	**97.11%**	**86.42%**	**90.42%**	**91.41%**

Table 1. Accuracy of the proposed LSTM+BERT model on Agency and Sociality classification.

References

[1] Byung-Chull Bae, Yun-Gyung Cheong, and Youngrok Song. "Modeling Happiness Using One-Class Autoencoder". In: (2019).

[2] Daniel Claeser. "Affective content classification using convolutional neural networks". In: *AffCon@ AAAI*. 2019.

[3] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[4] Kokil Jaidka et al. "The CL-Aff happiness shared task: Results and key insights". In: (2019).

[5] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[6] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[8] Matthew E Peters et al. "Deep contextualized word representations". In: *arXiv preprint arXiv:1802.05365* (2018).

[9] Lutz Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[10] Arun Rajendran, Chiyu Zhang, and Muhammad Abdul-Mageed. "Happy together: Learning and understanding appraisal from natural language". In: *arXiv preprint arXiv:1906.03677* (2019).

[11] Michael Saxon et al. "Word pair convolutional model for happy moment classification". In: (2019).

[12] Baohua Sun et al. "Squared english word: A method of generating glyph to use super characters for sentiment analysis". In: *arXiv preprint arXiv:1902.02160* (2019).

[13] Bakhtiyar Syed et al. "Ingredients for happiness: Modeling constructs via semi-supervised content driven inductive transfer learning". In: *Proceedings of the 2nd Workshop on Affective Content Analysis@ AAAI (AffCon2019), Honolulu, Hawaii (January 2019)*. 2019.

[14] Johnny Torres and Carmen Vaca. "CL-Aff Deep semisupervised clustering". In: (2019).

[15] Weizhao Xin and Diana Inkpen. "Happiness Ingredients Detection using Multi-Task Deep Learning". In: (2019).

Modeling Spatiosemantic Lateral Connectivity of Primary Visual Cortex in CNNs

Tonio Weidler, Mario Senden, and Kurt Driessens

Maastricht University, The Netherlands
{t.weidler,mario.senden,kurt.driessens}@maastrichtuniversity.nl

Contemporary computer vision frequently draws on convolutional neural networks (CNN). State-of-the-art performance is often achieved by further deepening previous architectures, but this increases computational costs and complicates the mapping of artificial layers to cortical areas in computational neuroscience, where these networks are used as models for goal-driven research. An alternative direction of multidisciplinary relevance is thus the search for structural and algorithmic improvements within or between layers to alleviate the necessity of additional depth. Inspiration for this can be found in visual cortex. Naturally, this also improves the network's biological plausibility, rendering it more useful for neuroscience.

In primary visual cortex, neurons project to subsequent visual areas but also connect laterally, i.e. to neurons in the same area [3]. Here, we distinguish three types of lateral interaction: *Semantic lateral connections* link neurons responding to the same patch of the visual field but preferring different line orientations. This type of connectivity typically follows a Mexican hat profile assumed to fine-tune the neurons' orientation selectivity [2]. *Spatial lateral connectivity* establishes interactions between neurons of similar orientation selectivity responding to different patches of the visual field along the axis of their orientation, presumably integrating and segmenting contours [3]. *Complex cells* receive input from phase selective simple cells and merge them into phase invariant representations [1]. In CNNs neither semantic nor spatial lateral connections are explicitly modeled and complex cells are only loosely captured by pooling.

We introduce a joint model of *spatiosemantic lateral connectivity* and an explicit model of complex cells to extend CNNs. Spatial and semantic lateral connections enrich the first convolutional layer by transforming its activation map with biologically inspired wavelets along both the spatial domain and the channel domains. We avoid the necessity of explicitly incorporating temporal dynamics resulting from recurrent interactions by assuming these dynamics to be linear. This allows us to solve for their steady state which renders the lateral connectivity a single non-parametric feedforward operation. Phase invariant complex cells are simulated by two independent cell populations s_a and s_b contributing their activations to the layer's representation individually, but additionally merging into a third population of complex cells via a pairwise complex modulus nonlinearity $\mathbf{c} = \sqrt{(\mathbf{s}_a + \mathbf{s}_b)^2}$. Unlike fixed complex wavelets in [4], the kernels of simple cells are learned autonomously. A full architecture of the adapted first convolutional layer is given in Figure 1.

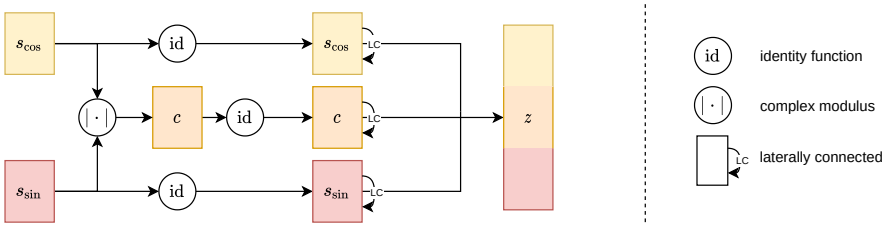


Fig. 1. Convolutional layer architecture with added complex cell simulation and lateral connections. s_a and s_b are two independent populations of neurons, realized as two arrays of convolutional filters. Their input is the original image, whereas the final output of the layer (z) constitutes the input to the second convolutional layer.

A qualitative analysis of the effects of our model of spatial lateral connectivity reveals that it successfully integrates segmented contours along straight lines. Experiments on object and texture classification showcase significant and substantial performance improvements in small-scale CNNs using complex cell simulations. Applied to texture classification, the combination of complex cells and spatial lateral connections produces the best performance, but spatial lateral connectivity on its own can already significantly improve a shallow network on both tasks. A closer look at the convolutional kernels emerging in laterally connected complex cells reveals their autonomously learned structure to be reminiscent of primary visual cortex. In particular, learned kernels largely adopt the orientation of their allocated spatial connectivity profiles and thus reinforce the proclaimed utility [3] of facilitation along this axis. In conclusion, our results demonstrate that introducing biologically inspired connectivity patterns into CNNs benefits their performance despite not increasing the number of trainable parameters. Improvements can be attributed to the integration and segmentation of contours during early visual processing, as the artificial connectivity profiles emulate those fulfilling these functions in the brain. In consequence, the introduced layer augmentation may not only improve small-scale CNNs in computer vision applications but also foster neuroscientific research relying on biologically plausible, goal-driven models.

References

1. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology **160**(1), 106–154 (Jan 1962)
2. Kang, K., Shelley, M., Sompolinsky, H.: Mexican hats and pinwheels in visual cortex. PNAS **100**(5), 2848–2853 (2003)
3. Lorenceau, J., Giersch, A., Series, P.: Dynamic competition between contour integration and contour segmentation probed with moving stimuli. Vision Research **45**(1), 103–116 (2005)
4. Mallat, S.: Group invariant scattering. Communications on Pure and Applied Mathematics **65**(10), 1331–1398 (2012)